

SUPERVISING CONSTRUCTE OF SOFTWARE

Patent number: CN1035375
Publication date: 1989-09-06
Inventor: YOHANSTON DAVY LANGLEY (US); NALO GEIMS LI (US); ROBOT HAYLON MARIOSEN-WIELENS (US)
Applicant: IBM (US)
Classification:
- international: **G06F9/44; G06F9/44;** (IPC1-7): G06F15/16
- european: G06F9/44G4
Application number: CN19880107805 19881114
Priority number(s): US19870122293 19871118

Also published as:

EP0317477 (A2)
US5237688 (A1)
JP1161531 (A)
EP0317477 (A3)
BR8806033 (A)

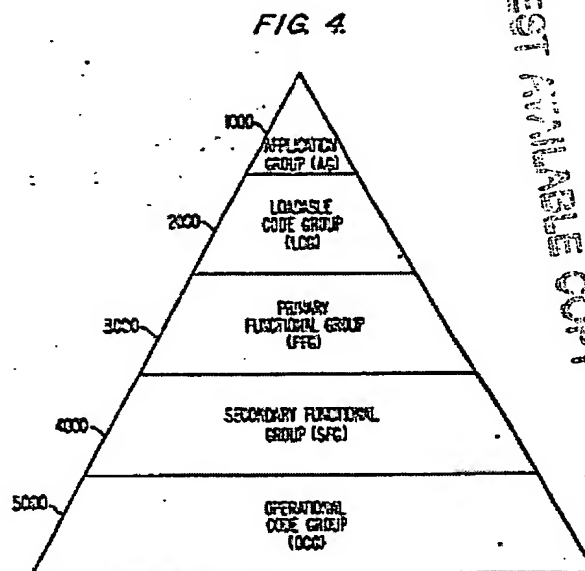
more >>

Report a data error he

Abstract not available for CN1035375

Abstract of corresponding document: **EP0317477**

Program developers use the program packaging tools of the invention to create programs packaged in four levels: LCG (2000), PFG (3000), SFG (4000), and OCG (5000). An application developer uses the application packaging tools of the invention to combine several of these program packages together to create an application package. This is done by adding the AG level (1000) on top of the four levels of the selected program packages. In addition, the application developer can select certain primary and secondary functions of the selected program packages while omitting other primary and secondary functions of the selected program packages. The packaging structure and packaging tools of the subject invention gives the developers the power and flexibility to create an application package tailor fit to meet the needs of a specific user.



Data supplied from the esp@cenet database - Worldwide



[12] 发明专利申请公开说明书

[21] 申请号 88107805.0

[51] Int.Cl⁴

G06F 15/16 —

[43] 公开日 1989年9月6日

[22] 申请日 88.11.14

[30] 优先权

[32] 87.11.18 [33] US [31] 122,293

[71] 申请人 国际商用机器公司

地址 美国纽约

[72] 发明人 戴维·郎利·约翰斯顿

詹姆斯·李·纳劳

海伦·玛丽奥森-威廉斯

罗伯特·哈利·沙蒂

丹尼斯·李·沙芬

加里·阿伯特·特克

[74] 专利代理机构 中国国际贸易促进委员会专利

代理部

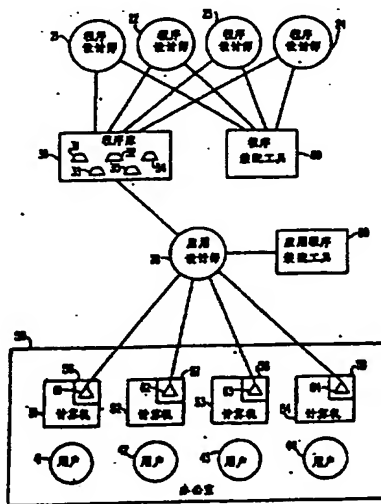
代理人 陈景峻

说明书页数: 22 附图页数: 33

[54] 发明名称 软件管理结构

[57] 摘要

一种软件管理结构, 应用程序包由几个联接的可替换单元组成。每个可替换单元均可在不影响其它可替换单元的情况下调用。这些可替换单元按多层结构方式上下联接在一起。程序设计师用本发明的程序装配工具按可装码组、主功能组、分功能组与操作码组四层装配制作程序。应用设计师在被选程序包的四层上面加一个应用组层, 应用程序装配工具将几个程序包组成一个应用程序包。此外, 应用设计师可以选择和删去被选程序包中的某些主分功能。



(BJ)第1456号

权 利 要 求 书

1. 计算机程序的多层软件管理结构，它包括：第一可替换单元与第二可替换单元，其特点是，所述第一可替换单元还包括第一首部与第二首部，所述第一首部包含第一标识符数据；

所述第二可替换单元还包括第二首部与第二体部，所述第二首部包含第二标识符数据；

所述第一体部包括所述第二标识符数据，因此允许所述第一可替换单元以多层结构方式与所述第二可替换单元相联。

2. 权利要求 1 提出的多层软件管理结构，其特点是：所述第一首部还包含第一硬件表，对于正常工作所述第一可替换单元与它有依附关系。

3. 权利要求 1 提出的多层软件管理结构，其特点是：所述第二首部还包含第二硬件表，对于正常工作所述第二可替换单元与它有依附关系。

4. 权利要求 1 提出的多层软件管理结构，其特点是：所述第一首部还包含第一软件表，对于正常工作所述第一可替换单元与它有依附关系。

5. 权利要求 1 提出的多层软件管理结构，其特点是：所述第二首部还包含第二软件表，对于正常工作所述第二可替换单元与它有依附关系。

6. 权利要求 1 提出的多层软件管理结构，其特点是：所述第一体部还包含一组对应于一组可替换单元的标识符信息。

7. 权利要求 1 提出的多层软件管理结构, 其特点是, 所述第二体部包含执行所述计算机程序所需的操作码。

8. 权利要求 1 提出的多层软件管理结构, 其特点是, 所述第一首部还包含第一维护信息。

9. 权利要求 2 提出的多层软件管理结构, 其特点是, 所述第二首部还包含第二维护信息。

10. 权利要求计算机层程序的多层软件管理结构, 它包括: 可装码组层与操作码组层, 其特点是: 所述可装码组层包含第二可替换单元,

所述操作码组层包含第五可替换单元,

所述第二可替换单元还包含第二首部与第二体部, 所述第二首部包含第二标识符数据,

所述第五可替换单元还包含第五首部与第五体部, 所述第五首部包含第五标识符数据。

11. 权利要求 10 提出的多层软件管理结构, 其特点是, 有一个包含第三可替换单元的主功能组层。

12. 权利要求 11 提出的多层软件管理结构, 其特点是有一个包含第四可替换层的分功能组层。

13. 权利要求 12 提出的多层软件管理结构, 其特点是有一个包含第一可替换单元的应用组层。

14. 权利要求 13 提出的多层软件管理结构, 其特点是, 所述第一可替换单元还包含第一首部与第二体部, 所述第一首部包含第一标识符信息。

15. 权利要求 14 提出的多层软件管理结构, 其特点是, 所述

第三可替换单元还包含第三首部与第三体部，所述第三首部包含第三标识符数据。

16. 权利要求15提出的多层软件管理结构，其特点是：所述第四可替换单元还包含第四首部与第四体部，所述第四首部包含第四标识符数据。

17. 权利要求16提出的多层软件管理结构，其特点是：所述第一体部包含所述第二标识符数据，因此允许所述第一可替换单元以多层结构方式与所述第二可替换单元相联。

18. 权利要求17提出的多层软件管理结构，其特点是：所述第二体部包含所述第三标识符数据，因此允许所述第二可替换单元以多层结构方式与所述第三可替换单元相联。

19. 权利要求18提出的多层软件管理结构，其特点是：所述第三体部包含所述第四标识符数据，因此允许所述第三可替换单元以多层结构方式与所述第四可替换单元相联。

20. 权利要求19提出的多层软件管理结构，其特点是：所述第四体部包含所述第五标识符数据，因此允许所述第四可替换单元以多层结构方式与所述第五可替换单元相联。

21. 权利要求20提出的多层软件管理结构，其特点是：所述第五体部包含执行所述计算机程序所需的操作码。

22. 编制具有一个主功能与一个分功能的程序包的方法，其特点是包括以下各步：

建立包括执行所述分功能所需的操作码与数据的第五层可替换单元；

为所述分功能建立第四层可替换单元；

将所述第四层可替换单元与所述第五层可替换单元相联；

为所述主功能建立第三层可替换单元；

将所述第三层可替换单元与所述第四层可替换单元相联；

建立第二层可替换单元；

将所述第二层可替换单元与所述第三层可替换单元相联；

23. 编制具有一组主功能与一组分功能的程序包的方法，其特点是包括以下各步：

建立一组包含完成一组所述分功能所需的操作码与数据的第五层可替换单元；

对于所述分功能组内的每个功能建立一个第四层可替换单元；

将所述第四层可替换单元与第五层可替换单元组的对应单元相联；

对于所述主功能组内的每个功能建立一个第三层可替换单元；

将所述第三层可替换单元与第四层可替换单元组的对应单元相联；

建立第二层可替换单元；

将所述第二层可替换单元与所述第三层可替换单元组相联。

24. 编制应用程序包的方法，其特点是包括以下各步：

选择具有一组主功能与一组分功能的第一程序包。

选择具有一组主功能与一组分功能的第二程序包。

将所述第一程序包与所述第二程序包合并，从所述第一与第二程序包选择某些将包括在所述应用程序包的所述主功能与分功能。

25. 编制应用程序包的方法，其特点是包括以下各步：

选择第一程序包，其第二层可替换单元与第三层可替换单元相

联。第三层可替换单元与第四层可替换单元相联，第四层可替换单元又与第五层可替换单元相联。所述第三层可替换单元对应于所述第一程序包的一个主功能，所述第四层可替换单元对应于所述第一程序包的一个分功能，所述第五层可替换单元包含执行所述第一程序包中所述分功能所需的操作码或数据。

选择第二程序包，其第二层可替换单元与第三层可替换单元相联，第三层可替换单元与第四层可替换单元相联，第四层可替换单元与第五层可替换单元相联。所述第三层可替换单元对应于所述第二程序包的一个主功能，所述第四层可替换单元对应于所述第二程序包的一个分功能，所述第五层可替换单元包含执行所述第二程序包中所述分功能所述的操作码或数据；

建立第一层可替换单元；

将所述第一层可替换单元与所述第一程序包的所述第二层可替换单元以及所述第二程序包的所述第二层可替换单元相联。

26. 编制应用软件包的方法，其特点是包括以下各步：

选择第一程序包。其第二层可替换单元与一组第三层可替换单元相联，第三层可替换单元组与第四层可替换单元组相联，第四层可替换单元组与第五层可替换单元组相联。每个所述第三层可替换单元对应于所述第一程序包的一个主功能，每个所述第四层可替换单元对应于所述第一程序包的一个分功能，所述第五层可替换单元组包括执行所述第一程序包中所述分功能所需的操作码或数据；

选择第二程序包。其第二层可替换单元与一组第三层可替换单元相联，第三层可替换单元组与第四层可替换单元组相联，第四层可替换单元组与第五层可替换单元组相联。每个所述第三层可替换单元对

应于所述第二程序包的一个主功能。每个所述第四层可替换单元对应于所述第二程序包的一个分功能，所述第五层可替换单元包括执行所述第二程序包中所述第二分功能所需的操作码或数据。

建立第一层可替换单元；

将所述第一层可替换单元与所述第一程序包的所述第二层可替换单元以及所述第二程序包的所述第二层可替换单元相联。

27. 权利要求26中提出的方法，其特点是还包括下一步：

从所述第一与第二程序包选择某些将包括在所述应用程序包中的所述主功能与分功能。

软件管理结构

本发明涉及数据处理领域。更具体地说，本发明是一种计算机软件包的装配结构，其中的软件程序由几个可替换单元按多层结构方式组合而成。

数据处理已发展到这样一个地步，用户希望他们的计算机能以最少的人为干预来处理它本身的资源。在处理计算机系统的硬件方面，已采用了自动布局与问题决定的方法，但有效地处理计算机系统的软件部分却还没有做什么工作。

当今典型的用户需要用几个不同的程序来实现他的特定应用。这些程序往往是由不同的程序设计师开发的。每个程序均与硬件和软件有关，因此用户或者应用程序设计师要将它们组成一个综合的系统并满足全部依附备件显得越来越难。此外，一个程序要能正确工作还可能依附于某一维护级或发售级的另外一个程序。通常用户的唯一办法是通过试凑或者利用大量的参考手册经过费力的探索来解决依附性的问题。

程序的结构往往与程序设计师的风格和思想方法有关。因此，两个程序具有一致结构，使一个程序能容易地发现另一个程序的信息，例如维护级或依附性信息，在目前是非常罕见的，即便同一个公司开发的两个软件也是如此。

目前程序设计师与用户面临的另一个问题是程序总是有些毛病需

要校正。通常程序设计师必须向每个用户送去一个全新的程序以校正差错，目前还没有一种有效的方法只改换出毛病的那一小部分代码。

现在，应用设计师的工作变得越来越复杂。当用户告知应用设计师他想引用一个应用软件包来满足他的特定需要时，他会感到要将几个不同的程序装配起来以满足用户的要求并保持今后扩充的灵活性有很多问题。通常，应用设计师不得不求诸于一堆昂贵的非结构式的通用程序，这些程序要么完成用户所不需的功能，要么不能够完成用户所需的全部功能。此外，一大批通用程序由于存贮了用户不需的程序，比实际需要白白多占用不少存贮器。对于那些存贮器已用完再无法在计算机内增存贮所需程序的用户，这个问题就更为棘手。

显然，程序设计师总是乐于使每个用户都对自己的程序称心如意。然而，在目前要开发专用程序来满足所有不同用户的要求，开发成本太高。程序设计师企图开发一种也许能满足所有用户要求的程序，这些用户有的需要程序的全部功能，有的只需要程序的某一功能。如果能做到这一点，程序设计师只需向每个用户提供用到的功能。当用户的需求增长时，再增加一些投资就可获得同一个程序中的附加功能。

本发明的主要目的是提出一种数据处理系统中管理程序的有效办法。

本发明的另一个目的是，提出一种计算机软件包的装配结构，其中应用软件包由几个可替换单元按多层次结构组合而成。

本发明的另一个目的是将应用软件包作为多层结构中的一个组成部分，这个多层结构能自动识别信息，维护级以及与硬件及其它软件的依附性。

本发明的另一个目的是提出一套软件装配工具，程序设计师可用它来设计程序模块，应用设计师可用它来设计应用软件包。

这些目的都是通过下述软件管理结构来达到的。

一个应用软件包由几个相联的可替换单元(RU)组成。每个可替换单元均可调用，而不影响其它可替换单元。这些可替换单元自上至下组成多层结构。在优选的实施方案中，一共有五层：应用组层(AG)、可装码组层(LCG)、主功能组层(PFG)、分功能组层(SFG)，以及操作码组层(OCG)。应用组层确定一组需要进行联接以完成能满足用户要求的高层应用的计算机程序。可装码组层确定一组完成某项一般工作的程序。主功能组层将可装码组层确定的公用程序细化，确定一组主功能。分功能组层将一次功能组层中确定的一次功能进一步细化，以确定一组能更接近于用户特定需要的专用分功能。操作码组层则包括由前面四层确定的运行用户专用应用程序的操作码。

每个应用组层可替换单元与一个或几个可装码组层可替换单元相联，后者又跟一个或几个主功能组层可替换单元相联，每个这种单元又与一个或几个分功能组层可替换单元相联，以下又与一个或几个操作码组层可替换单元相联。这样，就确定了一个五层的多层结构。

每个可替换单元由一个首部与体部组成。每个可替换单元的首部，不论处于哪一层，均包含自识别与维护信息，还可能包括一张与该单元有依附性的硬件与软件表。各层可替换单元的体部包含一张表格，列出了下一层中完成多层次链接的可替换单元，以此还有附加自识别与维护信息。可替换单元中的自识别、维护与依附性信息也称为重要产品数据(VPD)。操作码组层内每个可替换单元的体部则包

含运行程序所用的操作码。

应用软件的多层结构、可替换单元的可换性，以及这些单元内的目标别，维护与依附性信息，使软件变得更为完善。这种结构式软件可用来解决许多当前的数据处理问题，有些上面已讨论过。

程序设计师利用本发明的程序装配工具来编制程序，由四层装配而成；可装码组层、主功能组层、分功能组层和操作码组层。应用设计师利用本发明的应用程序装配工具将其中几个应用程序包装配在一起组成一个应用程序包。这是这样完成的：在所选层程序包的上面再加一个应用组层。此外，应用设计师还可以选择被选程序包中的某些主与分功能，删除被选程序包中的其它主功能与分功能。本发明提出的装配结构与装配工具给予应用设计师有力的手段与灵活性，来编制符合特定用户要求的应用程序包。

这种可装码多层结构的另一个优点是，不同类型的可装码，例如微码、购置的程序或用户自编应用程序，均可用一套公用的管理工具来管理，不需要对每类可装码采用一组特别的工具。因为每类代码的装配结构都相似，所以可这样做。

图 1 是本发明的环境。

图 2 是按本发明装配的软件库里的程序。

图 3 是装配起来满足特定用户要求的应用程序包。

图 4 示出了多层结构中的各层应用程序包。

图 5 是图 1 多层结构明细图，其中示出了可替换单元。

图 6 A 是组成应用组层、可装码组层、主功能组层与分功能组层的可替换单元的明细图。

图 6 B 是组成操作码组层的可替换单元的明细图。

图 7 是各可替换单元公用的固定首部的明细图。

图 8 是各可替换单元公用的可变首部的明细图。

图 9 是应用组层可替换单元中固定应用组层信息组的明细图。

图 10 是应用组层可替换单元中可变应用组层信息组的明细图。

图 11 是可装码组层可替换单元中固定可装码组层信息组的明细图。

图 12 是可装码组层可替换单元中可变可装码组层信息组的明细图。

图 13 是主功能组层可替换单元中固定主功能组层信息组的明细图。

图 14 是主功能组层可替换单元中可变主功能组层信息组的明细图。

图 15 是分功能组层可替换单元中固定分功能组层信息组的明细图。

图 16 是功能组层可替换单元中可变分功能组层信息组的明细图。

图 17 是程序设计师可使用的程序装配工具清单。

图 18~30 是定义程序装配工具与定义应用程序装配工具的明细图。

图 31 示出了采用显示程序装配工具的程序包实例。

图 32 是应用程序设计师可使用的应用程序装配工具清单。

图 33 示出了应用设计师怎样使用选择应用组功能工具的实例。

图 34 示出了采用显示应用装配工具的应用程序包实例。

图1是本发明的环境。程序设计师21-24按本发明的装配结构各自开发程序，以下将作较详细的说明。通常程序设计师21-24都是彼此独立工作的，且经常是互相竞争。尽管图1只示出了四名程序设计师，如果设计师都遵循本发明的装配结构，这些程序设计师可以是世界上几万名设计师。

由程序设计师21-24开发的程序放在程序库30。程序库30集中了程序设计师开发的全部程序。举几个例子，程序库30可能包括操作系统程序、文字处理程序、电子表格程序、游戏程序、教学程序、通讯程序和印刷程序。每个程序都完成主功能与分功能，以后将予以阐明。程序库30中程序31-35的明细图示于图2。

应用程序设计师26取出程序库30中的程序，重新将它们装配成一个应用程序包，以满足特定用户的要求。用户所需程序都装配在一个应用程序包内，将这个应用程序包交给用户。在用户得到的应用程序包中，只包括经选择的用户所需程序的一般功能与特定功能。

在图1中，应用程序设计师26分配到的工作是为办公室50中工作的用户41-44开发应用程序包。办公室50中有用户41-44使用的四台计师机51-54。计师机51-54的存贮区分别为56-59。存贮区56-59分别存放应用程序包61-64。虽然每个存贮区只表示出一个应用程序包，存放几个也是可以的。应用程序包61-64是由应用设计师26根据程序库30中由程序设计师21-24开发的程序编制的。应用程序包61-64分别能满足用户41-44的要求。

程序设计师21-24按本发明的装配结构用装配工具80装配他们的程序。同样，应用设计师26利用装配工具90重新将程序设

计师 21-24 编制的程序装配成一个能满足特定用户要求的应用程序包。装配工具 80 与 90 十分相似，在图 17~34 中将有更详细的说明。

图 2 是程序库 30 中程序 31-35 的明细图，请注意，程序 31-35 的图形象一个削去顶尖的金字塔。这个图形表示这些程序已按装配结构编好，但尚未成为应用程序包的一部分。

图 31 是字处理程序。这个特殊的字处理程序有两个主功能：文件准备与拼字检验。文件准备主功能有三个分功能：目录表功能、牵引生成功能以及虚拟打印功能。拼字检验主功能有三个分功能：常用字、法律字与医学用字。每个分功能相应的操作码放在 5000 层，用以完成该功能。

程序 32-35 是通讯程序、会计程序、操作系统与奕棋程序。请注意，这些程序除了奕棋程序 35 以外均有几个主功能与分功能。奕棋程序 35 很简单，只有一个主功能与分功能。

程序 31-35 仅仅表示程序设计师可使用装配工具 80 编制的程序。程序设计师有很大的灵活性来选择各种需要的主功能与分功能。他们选择的主功能与分功能越多，程序越是灵活，应用设计师越有可能在应用程序包中包括能满足特定用户需要的程序。

参阅图 1 与 2，假设办公室 50 是一个小型的法律事务所。应用设计师 26 与用户 41 谈话后得悉，用户 41 需要文字处理程序 31、会计程序 33 与操作系统 34，但不要通讯程序 32 或奕棋程序 35。应用设计师 26 还知道用户 41 需要文字处理程序 31 的文件准备与拼字校验主功能，以及目录来生成，通用字与法律用字分功能，但不需要任何其它的分功能。同样，还选择了会计程序 33 与操作系统

34的某些主分功能。按用户41需求，删除了某些功能。应用设计师26将程序31、33与34重新装配成一个应用程序包61，交给用户41，他将它存入自己计师机51的存贮器56。应用程序包62-64根据用户42-44的需求同样的方式编成。应用设计师26利用装配工具90编制应用程序包61-64，以下即将进一步叙述。

图3是应用设计师26为用户41开发的应用程序包61。请注意，应用程序包61的形状象一个已有顶尖的金字塔。应用程序包61只包括用户41需要的程序、主功能与分功能。如果用户41的需要有所改变，则可以为他开发一个新的应用程序包。

请注意，应用程序包61由五层组成：1000，2000，3000，4000与5000。图4-17更详细地图示了此优选实施方案的五层装配结构。

图4是优选实施方案的软件管理结构。请注意，共有五层，结构呈金字塔形以强调它的多层特性。最高层为应用组层（AG）1000，它确定了一组形成高层用户应用程序的计算机程序。举例来说，图3应用程序包61在应用组层确定一组法律事务所用的程序。第二层可装码组层（LCG）2000确定一组完成一般工作的程序。例如，图3的应用程序包61在可装码组层2000包括一个文字处理程序、会计程序与操作系统。第二层主功能组层（PFG）3000从可装码组层由确定的公用程序中进一步确定一组特定的主功能。例如，图3应用程序包61可装码组层内的文字处理程序，主功能组层包括文件准备与拼字校验主功能。其它程序在可装码组层也同样完成类似的细化工作。

第四层为分功能组层 (SFG) 4 0 0 0, 它将主功能组层中确定的主功能进一步细化为一组能更加满足用户需要的更为专用的分功能。分功能组层为每一个特定功能确定能装入计算机系统的最小可替换单元数。例如, 对于图 3 应用程序包 6 1 主功能组层内的拼字校验主功能, 分功能组层包括通用字与法律用字分功能。其它程序在主功能组层也可进行同样的细化工作。

最后一层操作码组层 (OCG) 5 0 0 0 包括在前面四层中确定的为运行专用的用户应用程序所需的操作码。

图 5 更详细地示出了图 4 优选实施方案的多层结构。请注意, 应用组层 1 0 0 0 由可替换单元 1 1 0 0 组成。可替换单元 1 1 0 0 的内容将在图 6 A、7 - 1 0 内作更详细的介绍。可替换单元 1 1 0 0 至少与可装码组层 5 0 0 0 内的一个可替换单元相联。如图 5 所示, 可替换单元 1 1 0 0 与可装码组层的可替换单元 2 1 0 0 相联。可替换单元 2 1 0 0 的内容在图 6 A、7、8、1 1 与 1 2 中有更详细的介绍。可替换单元 2 1 0 0 至少与主功能组层 3 0 0 0 内的一个可替换单元相联。如图 5 所示, 可替换单元 2 1 0 0 与主功能组层 3 0 0 0 内的可替换单元 3 1 0 0 相联。3 1 0 0 的内容在图 6 A、7、8、1 3 与 1 4 中有更详细的说明。可替换单元 3 1 0 0 至少与分功能组层 4 0 0 0 内的一个可替换单元相联。如图 5 所示, 可替换单元 3 1 0 0 与 4 1 0 0 相联。可替换单元 4 1 0 0 的内容在图 6 A、7、8、1 5 与 1 6 中有更详细的说明。可替换单元 4 1 0 0 至少与操作码组层内 5 0 0 0 的一个可替换单元相联。如图 5 所示, 可替换单元 4 1 0 0 与 5 6 0 0 相联。可替换单元 5 6 0 0 的内容在图 6 B、7 与 8 内有更详细的说明。

图6 A较详细地示出了组成应用组层、可装码组层、主功能组层、分功能组层的可替换单元。可替换单元1 0 0由首部1 1 0与体部1 2 0组成。首部1 1 0由固定首部2 0 0与可变首部3 0 0组成。体部1 2 0由固定组信息部4 0 0与可变组信息部5 0 0组成。固定首部2 0 0包括：长度信息2 2 0、可替换单元标识号信息2 3 0与偏移信息2 5 0。长度信息2 2 0记录可替换单元1 0 0的总长度以及可替换单元1 0 0体部1 2 0的长度。可替换单元标识符信息2 3 0包括自识别信息，它将本可替换单元与图5所有其它可替换单元区别开来。此外，可替换单元标识符信息2 3 0还包含可用来确定该单元维护级的维护信息。

偏移信息2 6 0包括对可变首部3 0 0与固定组信息部4 0 0的偏移数据。偏移信息还包括对于可变首部3 0 0中系统专用数据3 2 0、硬件依附性信息3 3 0以及软件依附性信息3 6 0的偏移数据。此外，偏移信息2 6 0还包括对固定组信息部4 0 0的偏移数据。图7较详细地介绍了固定首部2 0 0中的各项内容，图8较详细地介绍了可变首部3 0 0。

参阅图6 A，体部1 2 0由固定组信息部4 0 0与可变组信息部5 0 0组成。固定组信息部4 0 0包括体部长度信息4 2 0，组标识符信息4 3 0与组偏移信息4 6 0。体部长度信息记录了体部1 2 0的总长度。组标识符信息4 3 0包括附加的自识别信息与维护信息。组偏移信息4 6 0包括对可变组信息5 0 0的偏移信息。特别是，组偏移信息4 6 0还包括对组专用数据部5 2 0与下一层项目部5 4 0的偏移信息。

固定组信息4 0 0与可变组信息5 0 0内的具体内容决定于可替

换单元100是在应用组层1000、可装码组层2000，主功能组层3000还是在分功能组层4000。如果可替换单元100处于应用组层100，则固定应用组信息1400的详细内容如图9所示，可变应用组信息1500的详细内容如图10所示。如果可替换单元100处于可装码组层2000，则固定可装码组信息2400的详细内容如图11所示，可变可装码组信息2500的详细内容如图12所示。同样，如果可替换单元100处于主功能组层，则固定主功能组信息3400如图13所示，可变主功能组信息3500如图14所示。最后，如果可替换单元100处于分功能组层4000，则固定分功能组信息4400如图15所示，可变分功能组信息如图16所示。

现在来讨论图6B中的可替换单元600。可替换单元600只处于操作码组层5000。请注意，可替换单元600的首部110与图6A可替换单元100的首部110相同。可替换单元600还有一个体部620，它与可替换单元100的体部120不同。特别是，体部620包括操作码630。操作码630是非结构式的，它是程序的数据或可执行的代码。操作码630可以有选择地包含附加的重要产品数据（自标别、维护或依附性信息）。

前面在介绍图6A时已提到，可替换单元100的首部110不论可替换单元100处于哪一层均含有同样的内容。图7是固定首部200的明细图。长度信息220包括两段。信息段221确定该可替换单元的总长度。段222确定该单元体部的总长度。这个信息可用来链接各个可替换单元。标识信息230包括几段自识别信息与几段维护信息。可替换单元名称段231供系统作管理可替换单元之

用。各可替换单元的命名必须是唯一的，一些在系统中互相区别。图 3 1 中的 1 7 4 0 0 例示了一些可替换单元的名称。可替换单元访问标识符段 2 3 2 提供可替换单元的附加直观标识符信息。可替换单元装入标识符段 2 3 3 表明该单元是一种特殊装入操作，这是有些具有相应初始程序装入标识符的装置所需要的。这种机制允许系统决定哪些可替换单元应当在初始程序装入过程中装入。注意，可替换单元装入标识符段 2 3 3 是可选的。可装码组与应用组的标识符段 2 3 4、主功能组标识符段 2 3 6 以及次功能组标识符段 2 3 7 包含多层键结构中其它可替换单元的标识符。

现在来讨论固定首部 2 0 0 中标识符信息 2 3 0 中的维护信息段。可替换单元指令系统段 2 4 1 确定与可替换单元体部中的数据相兼容的机器指令系统。这一信息段也是任选的。可替换单元控制信息段 2 4 2 包括几个用以管理可替换单元的标志。其中之一是可替换单元修改标志，当单元被修改时置位。第二个标志是可替换单元的依附栏标志，如果该单元与任何硬件或软件有关，则置位。第三个标志是可替换单元类型标志，它确定该单元是属于应用组、主功能组、分功能组，还是操作码组。

可替换单元开发信息段 2 4 3 包括有关该单元原始开发的信息，诸如开发该可替换单元的某个研究所。可替换单元建立日期段 2 4 4 与建立时间段 2 4 6 包括建立此单元的日期与时间。可替换单元装入日期段 2 4 7 与装入时间段 2 4 8 包括该单元装入系统的日期与时间。可替换单元版本级段 2 4 9 与发售修改级段 2 5 0 规定了该可替换单元的维护级。可替换单元临时固定标识符段 2 5 2 与修补标识符段 2 5 3 都是任选的，它们提供该单元的临时标识符名称或修补标识

符名称。可替换单元公用名称段254是单元名称段231的二进制表示，可供系统软件选用。信息段257与258规定了有关可替换单元编释程序的信息。

现在来讨论组成固定首部200偏移信息段260的各段信息。对于单元体部段261的偏移信息指向该单元体部的起点。对于可替换单元系统专用数据段262的偏移信息指向单元可变首部中系统专用数据的起点。对于单元硬件依附性表段263的偏移信息指向单元可变首部中硬件依附性表的起点。对于可替换单元软件依附性表段264的偏移指向单元可变首部中软件依附性表的起点。信息段265指向单元专用信息的起点，段267则指向操作码组内可能出现的重要产品数据。

包含可替换单元的系统区各段266包括存放该单元的系统存储器中的地名。部分地名例示于图31的17600机。信息段270与271包括该单元最后一次修改的日期与时间。段272指向可变首部中的首部扩展数据。

图8是可变首部300中各信息段的明细图。系统专用数据320由系统专用数据长度段321与系统专用数据段322组成。

现在来讨论可变首部300硬件依附性表330中的各段信息。信息段331列出了为硬件依附表中的项目数。信息段332~337与339组成硬件依附表中的一个项目，并重复组成表中的各项。段332为可替换单元硬件相关标识符。几个不同的硬件，如果始终是一起使用的，可以给予同一个相关标识符。段333、335、336与337包括与该可替换单元有关的某种硬件的类型、型号、层次标识符号以及部品号。

现在讨论可变首部300软件依附性表360中的各段信息。信息段361为软件依附性表的容量，段362包含软件依附表的项目数。信息段363-376组成软件依附性表中的一项。这些段重复组成表中的其它各项。信息段363包含软件依附性表中该项的容量。段364包含软件依附性表中该项的软件相关标识符。几个不同的软件，如果始终是一起使用的，可给予同样的相关标识符。

这里假定，软件依附性表中所含的程序都是按本发明结构化的。因此，必须知道组成与此可替换单元有关的程序的各个可替换单元的多层链结构。可装码组标识符段367、可装码组主功能组标识符段370与主功能组-分功能组标识符段373提供确定这种多层链的信息。信息段368与369指出了依附程序可装码组的版本级与发售/修订级。同样，可装码组-主功能组的版本级与发售/修订级示于信息段371与372，主功能组-分功能组的版本级发售/修订级与中间程序变化级示于信息段374到376。信息段381-384组成首部扩展段。段382为可替换单元部品号。段383包括对该替换单元的简述。单元的简述例子示于图31的17050模。

现在来详述可替换单元100体部120中的各段信息。体部120如同前述，包含固定组信息400与可变组信息500。根据该单元处于多层链结构中的哪一层。信息400与500包含不同的信息段。可替换单元1100在应用组层1000有一组固定应用组信息1400，在图9中有较详细的说明。

长度信息1420有一个记录应用组体部总长度的信息段。现在来讨论可替换单元1100固定应用组信息1400中组成组标识符

信息1430的各段信息。信息段1431-1439为开发该应用组可替换单元的卖主名。卖主通常都是开发此应用程序包的应用设计师。信息段1441是一个特征常数，它提供一个直观标识符，表明该可替换单元是一个应用组型的单元。信息段1443-1446描述了该应用组可替换单元的标识符、部品号、版本级、发售/修订级。信息段1447与1448包含有关该应用组可替换单元各种中间或临时层间变动的信息。信息段1449包含可替换单元持主的版权信息。

现在来讨论组成可替换单元1100固定应用组信息1400中偏移信息1460的各段信息。信息段1463指向多层链中与此可替换单元相联的可装码组可替换单元表。图10较详细地示出了包含在可变应用组信息1500中的这张表格。段1464指向可变应用组信息1500中应用组专用信息的起点，详细内容示于图10。

现在来讨论组成图10中可变应用组可替换单元信息1500的各段信息。应用组专用数据1520包括信息段1521与1522，1521为应用组专用数据的长度，1522为应用组专用数据。

现在来讨论组成可装码组表1540的各段信息。信息段1541确定可装码组表的容量，段1542确定可装码组表的项目数。信息段1545-1556组成可装码组表中的一行，它们可重复组成可装码组表中的其它各项。信息段1545为选择/删除标志，当它置位时表示该可装码组已为应用组选中。信息段1546指明此可装码组是否能供应用组使用。信息段1547指出可用的可装码组是否已装入系统。信息段1548指示已装入的可装码组在此系统中已经调用。信息段1549-1548允许将可装码组作为应用程序包的一

部份提供给用户，但在用户提出扩展或升级之前不能调用。例如，会计出纳可装码组可以交给用户，但只有当用要扩大他的应用并需要些附加功能时才能调用。此信息段还可用作软件版权保护措施是一部分。

信息段 1 5 5 0 包含该可装码组在系统中的地址。系统地名的例子示于图 3 1 的 1 7 6 0 0 本。信息段 1 5 5 2 为可装码组的名称。系统用此来鉴别该可装码组。可替换单元名称的例子示于图 3 1 的 1 7 4 0 0 本。信息段 1 5 5 4 为可装码组的标识符。可装码组标识符的例子示于图 3 1 的 1 7 1 0 0 本。信息段 1 5 5 5 与 1 5 5 6 为可装码组各项的版本级与发售/修订级。

图 1 1 介绍了组成位于可装码组级 2 0 0 0 (图 5) 的可替换单元 2 1 0 0 中固定可装码组信息 2 4 0 0 的各段信息。请注意，其中不少信息段与图 9 所示固定应用组信息 1 4 0 0 中的信息段相似。请参阅图 9 关于这些段的介绍。信息段 2 4 3 1 - 2 4 3 9 中的卖主信息通常是指开发该程序包的程序设计师。信息段 2 4 6 3 指向与可装码组可替换单元相联的主功能组可替换单元来。这张表放在图 1 2 所示的可变可装码组信息 2 5 0 0 中。信息段 2 4 6 4 指向也包含在图 1 2 所示可变可装码组信息 2 5 0 0 中的可装码组专用信息。信息段 2 4 6 5 指向也包含在图 1 2 所示可变可装码组信息 2 5 0 0 中的应用组数据。信息段 2 4 6 7 指向信息 2 5 0 0 中的出口程序。

可替换单元 2 1 0 0 的可变可装码组信息 2 5 0 0 如图 1 2 所示。请注意，图 1 2 可变可装码组信息 2 5 0 0 中的许多信息段与图 1 0 中可变应用组信息 1 5 0 0 中的信息段相似。请参阅图 1 0 对这些信息段的说明。与图 1 1 信息段 2 4 6 5 有关的应用组数据示于图

12的信息段2531-2533。这些段内的信息使可装码组可以确定它要联接的应用组。出口程序的名称与地址示于信息段2561-2569。在编制应用程序包或程序包时，遇到差错就调用这些程序。

图13是主功能组层3000(图5)可替换单元3100中固定主功能组信息3400。请注意，图13固定主功能组信息3400中的某些信息段与图9固定应用组信息1400中的信息段相似。请参阅图9关于这些信息段的介绍。但应当注意，信息1400与2400中的卖主信息不再包含在信息3400之中。信息段3434确定多层链结构中此主功能组可替换单元的可装码组标识符。信息段3463指向与此主功能组可替换单元相联的分功能组可替换单元表。这张表放在可变主功能组信息3500之中，如图14所示。信息段3464指向放在图14可变主功能组信息3500中的主功能组专用信息。信息段3461与3462指向可能出现的放在可变主功能组信息3500中的硬件、软件依附性信息。

图14是3000层(图5)由可替换单元3100的可变主功能组信息3500。注意，图14信息3500中有些信息与图10信息1500中的信息段相同。请参阅图10关于这些段的说明。

分功能组层4000内可替换单元4100的固定分功能组信息4400(图5)示于图15。请注意，固定分功能组信息4400中许多信息段与图9固定应用组信息1400中信息段相同。请参阅图9关于这些段的说明。信息段4433与4434为该分功能组可替换单元确定多层链结构中更高层内的可替换单元。

信息段4461指向与此分功能组可替换单元相联的操作码组可

替换单元表。这张表放在可变分功能组信息 4 5 0 0 内，如图 1 6 所示。信息段 4 4 6 2 指向可变分功能组信息 4 5 0 0 中的临时固定/修补操作。信息段 4 4 6 3 指向可变分功能组信息 4 5 0 0 中的分功能组专用信息。

可替换单元 4 1 0 0 的可变分功能信息 4 5 0 0 如图 1 6 所示。请注意，图 5 信息 4 5 0 0 中的有些信息段与图 1 0 信息 1 5 0 0 中的信息段相似。请参阅图 1 0 关于这些段的说明。信息段 4 5 6 1 - 4 5 7 0 确定与此可替换单元有关的临时固定/修补操作表。

现在回到图 1。讨论程序设计师 2 1 - 2 4 用来开发应用程序包的程序装配工具 8 0。程序装配工具 8 0 是一种能在象个人计算机这类通用计算机上运行的软件，它是经过精心设计的，如图 1 7 - 3 4 所示。图 1 7 是程序设计师能用于编制程序包的工具清单。程序设计师利用图 1 7 所列的程序装配工具，开发按可装码组、主功能组、分功能组与操作码组四层结构装配的程序包。程序装配工具 1 1 0 0 0 允许程序设计师确定程序包。装配包 1 1 0 0 0 的详细情况示于图 1 8 - 3 0。装配工具 1 7 0 0 0 允许程序设计师显示由装配工具 1 1 0 0 0 确定的程序包。显示给程序设计师的程序包的例子示于图 3 1。

图 1 8 是装配工具 1 1 0 0 0 的总控流程。注意，图 1 8 还同时示出了确定应用程序包的装配工具 2 1 0 0 0 的总控流程，这是因为这两种工具很相似。在图 1 8 - 3 0 的流程图中，应用程序装配工具 2 1 0 0 0 所独有的部分的参照编号为 2 1 0 0 0 - 2 1 9 9 9。这些部分以后在讨论应用程序装配工具 2 1 0 0 0 时再说明。决定块 1 1 0 1 0 查问程序设计师，他是否要确定一个可装码组。如果程序

设计师表示，他想要确定一个可装码组，则在信息块11100中确定可装码组的体部。信息块11100的详细情况示于图190接着在信息块11200中确定可装码组首部。信息块11200的详细内容示于图25。决定块11020询问程序设计师他是否想确定一个主功能组。如果想确定，则在信息块11400中确定主功能组体部。信息块11400的详细内容示于图26。接着，在信息块11200中确定了主功能组首部。信息块11200的详细内容示于图25。请注意，信息块11200对于确定可装码组、主功能组、次功能组、应用组与操作码组层的首部都是公用的。

决定块11030询问程序设计师，他是否要确定一个分功能组。如果想确定，则在信息块11500中确定次功能组体部。信息块11500的详细情况示于图27。接着，如同上述。在信息块11200中确定分功能组首部。决定块11050询问程序设计师，他是否要确定一个操作码组。如果是，则信息块11060则联通包含程序操作码与数据的操作码组体部。如同上述。信息块11200定义操作码组首部。信息块11070询问程序设计师，是否已完成用工具11000的开发工作。注意，如果程序设计师正在从头开始设计一个程序包，他可能先确定操作码组，接着是分功能组，然后是主功能组，最后是可装码组。

图19说明图18中的信息块11100怎样确定可装码组体部。信息块11110向程序设计师提示图11所示信息段2431-2439中的卖主信息。通常这些信息段的卖主就是程序设计师本人，或者是他工作的公司，当然也可以是其他卖主。信息块11110要求程序设计师为可装码组选择特征常数。这个数据放在图11信息

段2440内。信息块11111提问程序设计师关于可装码组标识符与部品号的信息。这个信息位于图11信息段2450与2451内。信息块11112提问程序设计师关于可装码组维护级与版权的信息。此信息位于图11的信息段2452-2456。信息块11120确定与此可装码组有关的主功能组表。信息块11120的详细内容示于图20，以后还将讨论。信息块21113与21130是应用设计师用的，放在以后讨论。

决定块11114查问程序设计师是否需要可装码组专用信息。如果需要，则在信息块11140确定可装码组专用数据。信息块11140的详细情况示于图22，以后再说明。信息块11115查问程序设计师是否有与此可装码组有依附性的硬件。如果有，则在信息块11160中确定可装码组硬件依附性表。信息块11160的详细情况示于图23，以后再讨论。

决定块11116查问程序设计师是否有与此可装码组有依附性的软件。如果有，则在信息块11180中确定可装码组软件依附性表。信息块11180的详细情况示于图24，以后还将讨论。在图19中信息块11117保存了程序设计师确定的可装码组体部的定义。在流程图中信息块11118进入信息块11200，在此确定可装码组首部。以后将配合图25对信息块11200作更详细的说明。

图20较详细地说明了如何在信息块11120中定义主功能组表。信息块11121查问程序设计师关于主功能组名称与标识符的信息。这个信息放在图12的信息段2550与2552内。信息块11122查问程序设计师关于主功能组维护级的信息。此信息放在

图12信息段2553与2454内。信息块11124询问用户在此表内可要确定其它的主功能组。如果不要，则重复信息块11121-11123，直到表中的全部主功能组均被确定为止。当表中全部主功能组被确定之后，信息块11125设定图12中主功能组表容量段2541与项目数段2543。信息块11126回到图19的信息块21113。

图22详细说明了信息块11140怎样确定可装码组专用数据。信息块11141询问程序设计师关于可装码组专用数据的信息。可装码组专用数据放在图12信息段2522内。信息块11142根据信息段2522的内容确定可装码组专用数据的长度。此长度放在图12的信息段2521内。信息块11143回到图19的信息块11115。

图23较详细地说明了信息块11160怎样确定可装码组硬件依附性表。决定块11161询问程序设计师对于此硬件依附性表是否需要一个硬件相关标识符。如果需要，则硬件相关标识符放在图8信息段332内，如信息块11162所示。信息块11163提示给程序设计师输入硬件的类型与型号。此信息放在图8的信息段333与335内。决定块11164询问程序设计师对于硬件依附性表中的这一项目是否需要装入标识符。如果需要，硬件装入标识符放在图8的信息段336内。如信息块11165所示。如果不需要，则跳过信息块11165。信息块11166提示给程序设计师输入硬件的部品号。这个信息位于图8的信息段337。决定块11167询问是否还有更多的硬件依附性要填入此表。如果还有，则重复信息块11161-11166，直到表中的全部硬件依附性

填入为止。当全部硬件依附性填入此表后，信息块11168修改图11中硬件依附性表容量信息段338与项目数信息段331，加入新的信息。信息块11169回到图19的信息块11116。

图24详细说明了信息块11180怎样确定可装码组软件依附性表。信息块11181询问程序设计师此软件依附性表是否需要一个软件相关标识符。如果需要，则软件相关标识符放在图8信息段364内，如信息块11182所示。如不需要，则跳过信息块11182。信息块11183询问程序设计师关于软件依附性类型的信息。此信息位于图8信息段366内。信息块11184询问程序设计师有关相依附软件的可装码组标识符与维护级。此信息放在图8的信息段367~369。信息块11185询问用户有相依附性的软件是否是分功能组可替换单元。如果是，则在信息块11186向程序设计师提示分功能组标识符与维护级，此信息位于图8的信息段373-375内。

决定块11187询问对分功能组中间程序变化层是否有依附性。如果有，信息块11188向程序设计师提示输入次功能组程序变化中间标识符。此信息位于图8的信息段376。如果没有，跳过信息块11188。信息块11189询问程序设计师对主功能组可替换单元是否有依附性。如果有，则向程序设计师提示输入主功能组标识符与维护级。此信息位于图8的信息段370-372，如信息块11190所示。如果没有，则跳过信息块11190。信息块11191询问程序设计师是否所有软件依附性均已确定。如果没有，控制回到图24的起点，重复信息块11181-11190，直到全部软件依附性确定为止。一旦全部软件依附性被确定，信息块

1 1 1 9 2 就修改图 8 中软件依附性容量信息段 3 6 1 与项目数信息段 3 6 2。信息块 1 1 1 9 3 回到图 1 9 的信息块 1 1 1 7 7。

图 2 5 A - C 详细说明了图 1 8 中的信息块 1 1 2 0 0 如何确定可装码组首部。注意,可装码组首部的确定方式与图 1 8 中确定主功能首部、次功能组首部、应用组首部或操作码组首部是一样的。信息块 1 1 2 1 1 向程序设计师提示输入可替换单元的名称与访问标识符。此信息位于图 7 的信息段 2 3 1 与 2 3 2。信息块 1 1 2 1 2 向程序设计师提示输入可替换单元的版本级与发售级信息。此信息位于图 7 的信息段 2 4 9 与 2 5 0。信息块 1 1 2 1 3 向程序设计师提示输入可替换单元的控制信息与开发信息。此信息位于图 7 的信息段 2 4 2 与 2 4 3。信息块 1 1 2 1 4 设定存放此可替换单元的系统区的名称。此信息位于图 7 的信息段 2 6 6。信息块 1 1 2 1 5 设定此可替换单元的开发日期与时间。此信息位于图 7 的信息段 2 4 4 与 2 4 6。

信息块 1 1 2 1 9 查问此可替换单元是否是可装码组可替换单元。如果是,则信息块 1 1 2 2 0 设定图 7 信息段 2 3 4 中的可装码组标识符。信息块 1 1 2 2 1 对信息块 1 1 1 0 0 确定的可装码组体部进行检索。

信息块 1 1 2 2 2 查问此可替换单元是否是主功能组可替换单元。如果是,则在图 7 的信息段 2 3 4 与 2 3 6 设定可装码组标识符与主功能组标识符,如信息块 1 1 2 2 3 所示。在信息块 1 1 4 0 0 确定的主功能组体部在信息块 1 1 2 2 4 内进行检索。

信息块 1 1 2 2 5 查问此可替换单元是否是分主能组可替换单元。如果是,信息块 1 1 2 2 6 设定图 7 信息段 2 3 4、2 3 6 与

237中与此分功能组有关的可装码组、主功能组与分功能组标识符。

接着,信息块11255获得信息块11500中确定的分功能组体部,信息块11227(图25B)查问对此可替换单元硬件依附性表是否已经确定。如果是,信息块11228决定硬件依附性偏移信息是否在可替换单元的体部或首部。可替换单元依附性偏移信息,在此优选实施方案中,可以在可替换单元的体部,也可在可替换单元的首部,但不能同时存在。如果硬件依附性偏移表位于可替换单元体部,则在信息块15230中将硬件依附性表附加到可替换单元的体部。如果硬件依附性偏移表位于可替换单元的首部,则在信息块11229中将硬件依附性表附加到可替换单元的首部。在后一种情况下,信息位于图7的信息段263,向硬件依附性表提供偏移信息。

信息块11231查问可替换单元的软件依附性表是否已确定。如果是,则信息块11232查问软件依附性偏移信息位于可替换单元的体部还是首部。如果软件依附性偏移位于可替换单元的体部,则在信息块11234中将软件依附性表附加到可替换单元的体部。如果软件依附性表偏移信息位于可替换单元的首部,则在信息块11233中将软件依附性表附加到可替换单元的首部。与此偏移有关的信息位于图7的信息段264。

信息块11235查问此可替换单元的系统专用数据是否已确定。如果已确定,则信息块11236查问该系统专用数据的偏移信息是否包含在可替换单元的体部或首部。如果系统专用数据的偏移信息位于可替换单元的体部,则在信息块11238中将此系统专用数

据送入可替换单元的体部。如果系统专用数据偏移信息位于可替换单元的首部，则在信息块1 1 2 3 7中将系统专用数据送入可替换单元的首部。与偏移有关的数据位于图7信息段2 6 2。

信息块1 1 2 3 9查问此可替换单元的可替换单元专用信息是否确定。如果已确定，则信息块1 1 2 4 0查问此可替换单元专用信息的偏移信息是否包含在可替换单元的体部或首部。如果可替换单元专用信息偏移位于可替换单元的体部，则在信息块1 1 2 4 2中将可替换单元专用数据送入可替换单元的体部。如果可替换单元专用信息偏移位于可替换单元的首部，则在信息块1 1 2 4 1中将可替换单元专用信息送入可替换单元的首部。有关偏移的信息存放在图7的信息段2 6 5。

信息块1 1 2 4 3（图2 5 C）查问此可替换单元是否是操作码组可替换单元。如果是，则信息块1 1 2 4 4设定该操作码组的可装码组、主功能组与分功能标识符。此信息位于图7的信息段2 3 4、2 3 6与2 3 7。信息块1 1 2 4 5查问此操作码组是否有装入标识符。如果有，则在信息块1 1 2 4 6中将操作码组装入标识符送入图7的信息段2 3 3。

信息块1 1 2 4 7查问此操作码组是否有一个公用名称。如果有，将此操作码的公用名称放入图7的信息段2 5 4，如信息块1 1 2 4 8所示。信息块1 1 2 4 9查问操作码组体部内是否包含重要产品数据。如果有，将图7的信息段2 6 7加以修正，将偏移放入操作码体部中含有重要产品数据的部分，如信息块1 1 2 5 0所示。信息块1 1 2 5 1将图7信息段2 2 2内的可替换单元体部总长度加以修正。信息块1 1 2 5 2对可替换单元的体部长度加以修正，如果

它是应用组可替换单元,则该长度位于图9的信息段1420;如果它是可装码组单元,则位于图11的信息段2420;如果它是主功能组单元,则在图13的信息段3420;如果它是分功能组单元,则位于图15的信息段4420。信息块11253计算可替换单元的总长度,将此信息放入图7的信息段221。可替换单元的总长度包括首部与体部的长度。信息块11254回到信息块11020。

图26较详细地说明了图18的信息块11400怎样确定主功能组体部。信息块11410提示程序设计师输入此主功能组的标识符以及获得此主功能组的可装码组标识符。此信息位于图13的信息段3432与3434。信息块11411提示程序设计师输入主功能组部品号、维护级与版权信息。此信息位于图13的信息段3435-3450。信息块11412查问程序设计师此主功能组是否与任何其它硬件有依附性。如果有,则确定硬件依附性表,如信息块11160所示。信息块11160的详细情况示于图23,上面已讨论过。信息块11413查问程序设计师此主功能组是否存在软件依附性。如果回答是肯定的,则信息块11180确定此主功能组的软件依附性。信息块11180的详细情况示于图24,以上已讨论过。信息块11120定义与此主功能组相联的包括全部与功能组的分功能组表。信息块11120的详细情况示于图20,上面已讨论过。信息块11414查问程序设计师是否需要主功能组专用信息。如果需要,则在信息块11140中确定主功能组专用信息。信息块11140的详细情况示于图22,以上已讨论过。主功能组专用信息的偏移位于图13的信息段3464。信息块11415保存图26中确定的主功能组体部。如图25所示,信息块11416随流

程图进入信息块11200,在此确定功能组首部。

图27详细示出了信息块11500怎样定义图18中的分功能组体部。信息11510提示程序设计师输入分功能组标识符以及获得此分功能组的可装码组、主功能组的标识符。此信息位于图15的信息段4431、4433与4434。信息块11511提示程序设计师输入分功能组部品号、维护级以及版权信息。此信息位于图

15的信息段4435-4440。信息块11512查问程序设计师此分功能组是否依附于任何其它硬件。如果回答是肯定的,则在信息块11160确定分功能组硬件依附性表。信息块11160的详细情况示于图23,上面已予讨论。信息块11513查问程序设计师此分功能组是否存在软件依附性。如果存在,在信息块11180确定此分功能组的软件依附性表。信息块11180的详细情况示于图24,以上已予讨论。信息块11520建立将此分功能组与其它各操作码组相联的操作码组表。信息块11520如何建立此操作码表的详细情况示于图28,以后还将说明。此操作码组的偏移信息位于图15中信息段4461。信息块11550为此分功能组建立分功能组临时固定/修补操作表。这张表放在图16的信息段4561-4570。此表的偏移信息放在图15的信息段4462。信息块11514查问是否需要任何分功能组专用信息。如果回答是肯定的,则在信息块11140建立分功能组专用信息。图11140的详细情况示于图22,以上已予说明。分功能级组专用信息放在图

15的信息段4521,分功能组专用信息长度放在信息段4520,分功能组专用信息的偏移放在信息段4463。信息块11512保留在图27中确定的分功能组体部。信息块11516继续随控制流

程进入信息块 1 1 2 0 0，在此确定分功能组首部。

图 2 8 详细说明了信息块 1 1 5 2 0 怎样确定操作码组表。信息块 1 1 5 2 1 提示程序设计师输入包含此操作码组的系统地名。此信息位于图 1 6 的 4 5 4 4。信息块 1 1 5 2 2 获得标志位于系统某一地址的操作码组的变量。这一点可以这样完成：查问某一地址中公有某一名称的全部操作码组，或者用人工将这些操作码组输入。

信息块 1 1 5 2 3 查问此操作码的名称以及操作码组的付标识符。信息块 1 1 5 2 4 查问此操作码组是否属于另一个与功能组。如果回答肯定，则在信息块 1 1 5 2 5 中修改异常操作码组表。一个操作码只能属于一个分功能组。信息块 1 1 5 2 6 将操作码组的名字及付标识符放入图 1 6 的信息段 4 5 4 7 与 4 5 4 8。信息块 1 1 5 2 7 查问在系统某一地址是否存在此操作码组。如果不存在，则信息块 1 1 5 2 8 使维护级信息等于获得此操作组的与分功能组的维护级信息。因此，图 1 5 信息段 4 4 3 6 中的信息被制入图 1 6 信息段 4 5 4 9，图 1 5 信息段 4 4 3 7 中的维护信息被制入图 1 6 的信息段 4 4 5 0。

信息块 1 1 5 2 9 查问操作码组维护级信息。此信息位于信息段 4 5 4 9 - 4 5 5 4。信息块 1 1 5 3 0 查问系统地址中的全部操作码组是否已全部处理完毕。如果还没有，控制这回到信息块 1 1 5 2。如果已将它所全部处理完毕，则信息块 1 1 5 3 1 设定图 1 6 信息段 4 5 4 3 中系统地址表的容量，以及图 1 6 信息段 4 5 4 6 中表内的操作码组项目数。信息块 1 1 5 3 2 查问是否全部系统地址已处理完毕。如果没有，控制这回到信息块 1 1 5 2 1，重复进行。如果全部系统地址已处理完毕，信息块 1 1 5 3 3 设定图 1 6 信息段 4 5 4 1

中操作码组表的容量，以及图 16 信息段 4 5 4 2 中系统地址的项目数。信息块 1 1 5 3 4 将不制表的操作码组告知用户。信息块

- 4 1 1 5 3 5 将控制这回到图 27 的信息块 1 1 5 5 0。当程序设计师确定一个或几个程序包后，他就可以利用装配工具 1 7 0 0 0 将这些程序包在他的显示器上显示出来。例如，如果程序设计师 2 1 利用确定程序包工具 1 1 0 0 0 已确定了图 2 程序仓 3 0 中的五个程序包，则程序设计师能见到的显示内容如同图 3 1 所示。1 7 0 5 0 概的显示为简单的说明。可装码组标识符显示在 1 7 1 0 0 概。如果有主功能组标识符，则显示在 1 7 2 0 0 概。如果有分功能组标识符，则显示在 1 7 3 0 0 概。可替换单元名称显示在 1 7 4 0 0 概。可替换单元类型显示在 1 7 5 0 0 概。存放可替换单元的系统地名显示在 1 7 6 0 0 概。1 7 7 0 0、1 7 8 0 0 与 1 7 9 0 0 概分别显示可替换单元的可用标志，已装标志以及选用标志。在用户使用可替换单元的功能之前，必须先将这三个标志置位。注意，有些可替换单元只是可用的，有些是可用并已安装的，有些则是可用、已安装并选用的。

再回到图 1，应用设计师 2 6 可利用应用程序装配工具 9 0 来编制应用软件包。应用设计师选择程序库 3 0 中的某些程序包 概成一个应用程序包。此外，选择了这些程序包中的某些主分功能，删除了某些功能。应用设计师 2 6 利用应用程序装配工具 9 0，编制一个包括应用组、可装码组、主功能组、分功能组与操作码组五层的应用程序，来完成上述工作。应用程序装置工具 9 0 包括一组软件，它们能够在诸如个人计算机这一类的通用计算机上运行，如图 17 - 3 4 所示，它们是经过精心设计的。

图 3 2 是显示屏上的一张清单，它列出了应用设计师可用开发应用程序包的各种程序装配工具。装配工具 2 1 0 0 0 确定一个应用程序包。装配工具 2 4 0 0 0 选择应用程序包中每个程序包的主分功能。装配工具 2 4 0 0 0 显示已编成的应用程序包。装配工具 2 1 0 0 0 与上述装配工具 1 1 0 0 0 十分相似。除了工具 1 1 0 0 0 的功能以外，装配工具 2 1 0 0 0 还有少数几个供应用设计师使用的功能，通常对程序设计师是没有的。这些附加的功能在图 1 8 - 3 0 中用参照号 2 1 0 0 0 - 2 1 9 9 9 表示，下面将讨论之。

首先请参阅图 1 8，决定块 2 1 0 4 0 查问程序设计师他是否要确定一个应用组。如果需要，则在信息块 2 1 6 0 0 中确定应用组体部。图 2 9 较详细地说明了信息块 2 1 6 0 0。如前面所述，在信息块 1 1 2 0 0 里进一步确定应用组首部。

现在，参阅图 1 9，信息块 2 1 1 1 3 查问应用设计师此可装码组是否需要应用组信息。如果要，则在信息块 2 1 1 3 0 中确定一张应用组表。信息块 2 1 1 3 0 的详细情况示于图 2 1，以下将作说明。

信息块 2 1 1 3 0 确定一张应用组表，较详细的情况示于图 2 1。信息块 2 1 1 3 1 提示应用程序设计师输入应用组的标识符。此信息位于图 1 2 的信息段 2 5 3 1。信息块 2 1 1 3 2 提示应用设计师输入应用组的名称。此信息位于图 1 2 的信息段 2 5 3 2。信息段 2 1 1 3 3 提示应用设计师输入存放应用组可替换单元的系统存贮器地址。此信息位于图 1 2 的信息段 2 5 3 3。信息块 2 1 1 3 6 返回到图 1 9 的信息块 2 1 1 1 5。

现在请参阅图 2 5 A，信息块 2 1 2 1 6 查问此可替换单元是否

为应用组单元。如果是，则在图7信息段234内设定该可替换单元的应用组标识符，如信息块21217所示。信息块21218对信息块21600中确定的应用组体部进行检索，如图2.9所示。

图2.9较详细地说明了信息块21600怎样定义应用组体部。信息块21610提示应用设计师输入应用组的卖主信息与特征常数。此信息位于图9的信息段1431-1439。信息块21611提示应用设计师输入应用组标识符与应用组部品号。此信息位于图9的信息段1443与1444。信息块21612提示应用设计师输入应用组维护与版权信息。此信息位于图9的信息段1445-1449。信息块21620建立包括与此应用组相联的全部可装码组的可装码组表。信息块21620怎样建立可装码组表的详细过程示于图30，以后还将讨论。信息组21613提问程序设计师，是否需要任何应用组专用信息。如果需要，在信息块11140中确定应用组的专用信息。信息块11140怎样确定应用组专用数据的详细过程示于图2.2，已给予以讨论。信息21614将图2.9中确定的应用组体部存入存储器。信息块11615继续随控制流程进入信息块11200，在此确定应用组首部。信息块11200怎样确定应用组首部的详细过程示于图2.5，已予以讨论。

信息块21620怎样确定可装码组表的详细过程示于图30。信息块21621向程序设计师提示输入可装码组的名称与标识符。此信息位于图10信息段1552与1554。信息块21622查问程序设计师关于可装码组维护级的信息。此信息位于图10的信息段1555与1556。信息块21623设定可装码组可用性信息。并向程序设计师提示存放该可装码组系统地址。此信息位于图

10信息段1546与1550。信息块21624查询是否表格中的全部可装码组已确定。如果还没有，则重复进行信息块21621-21623。当全部可装码组被确定后，信息块21625将可装码组的项目数放入图10的信息段1541，将可装码组表的容量放入图10的信息段1543。信息块21626返回到图29的信息块21613。

应用设计师利用工具21000选定程序包所含的可装码组确定应用程序包后，设计师接着就利用工具24000选择应用程序包功能。例如，假定应用设计师26要为用户41开发一个应用程序包61，如图1所示，则图33将显示出被选程序包的清单。注意，24050-24900框与31中17050-17900框相同。24010框为应用设计师提供一个选择应用程序包所需主功能与分功能的场合。为了确定图3应用程序包61，应用设计师对于每个图3中被选的主功能与分功能要在24010框记入。在应用程序包交给用户之前，不被选中的主功能与分功能从可装码组与主功能组的体部删去。

装配工具27000显示用工具21000与24000编制的应用程序包。图34是程序设计师编完图3的应用程序包61后能看到的显示。

以上已讨论过，优选实施方案中程序包含五层，如图4所示。但是，在本发明范围内，并非所有这些层都是必需的。曾经考虑另一种实施方案，其中程序只有两层：可装码组层2000与操作码组层5000。虽然这种实施方案不能象优选实施方案的五层结构能满足用户的需求，这种实施方案的许多优点是十分明显的。在这种实施方

案中，没有图9、10、13、14、15与16。图11的信息段2463变为“操作码组表偏移信息”，信息段2465变为“保留”。同样，图12所示的主功能组表为操作码组表所替代，同时选择/删除段2544变为“保留”。在此实施方案中没有信息段2531-2533。同样，要确定只有两层的程序包图18-30流程图中有不少部分是不需要的。

虽然本发明以上所述只涉及优选实施方案与另一种方案，但专家都了解，在不偏离本发明的精神、范围与要义的前提下，在细节上可以有許多变化。因此，这里讨论的只局限于下述要求中所指出的内容。

图 1

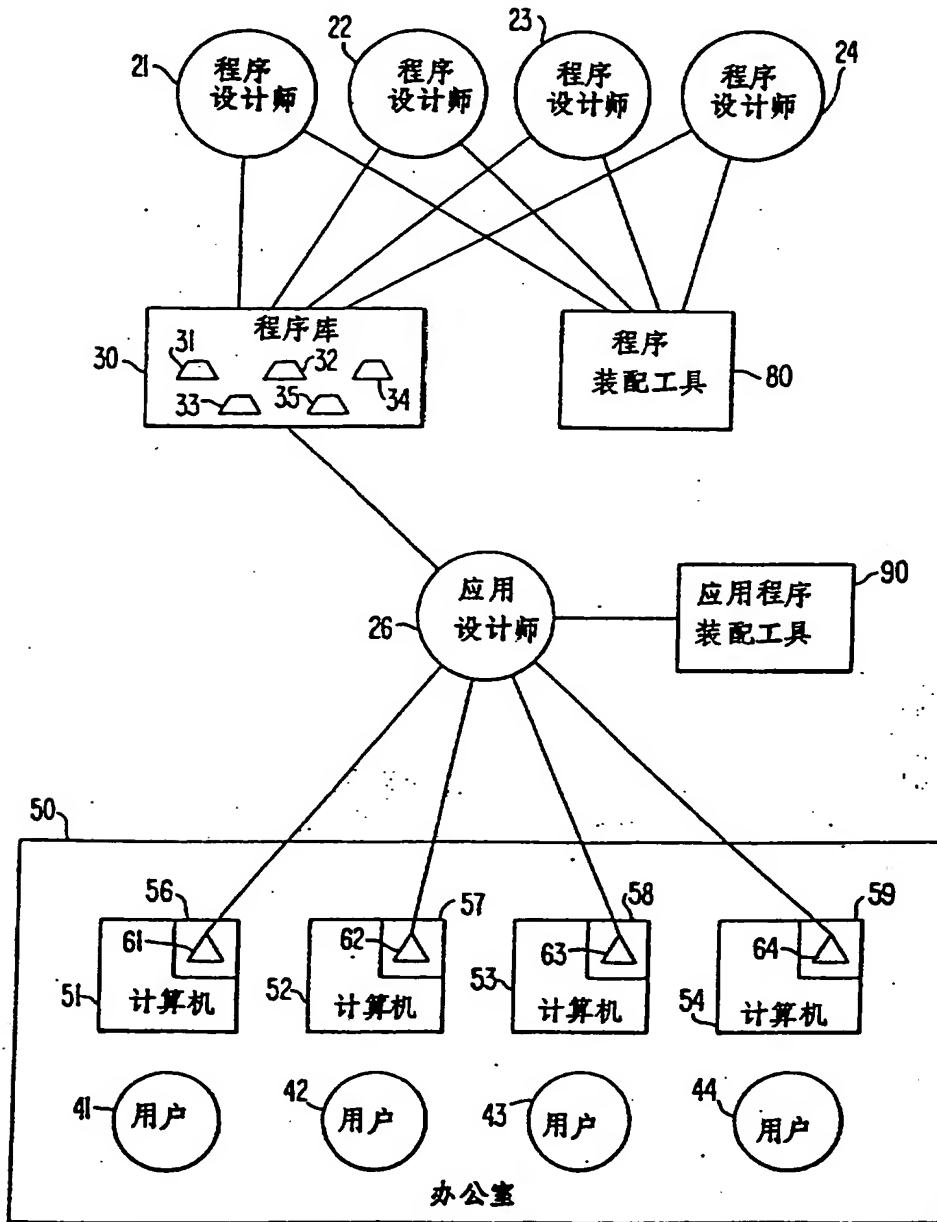


图 24.

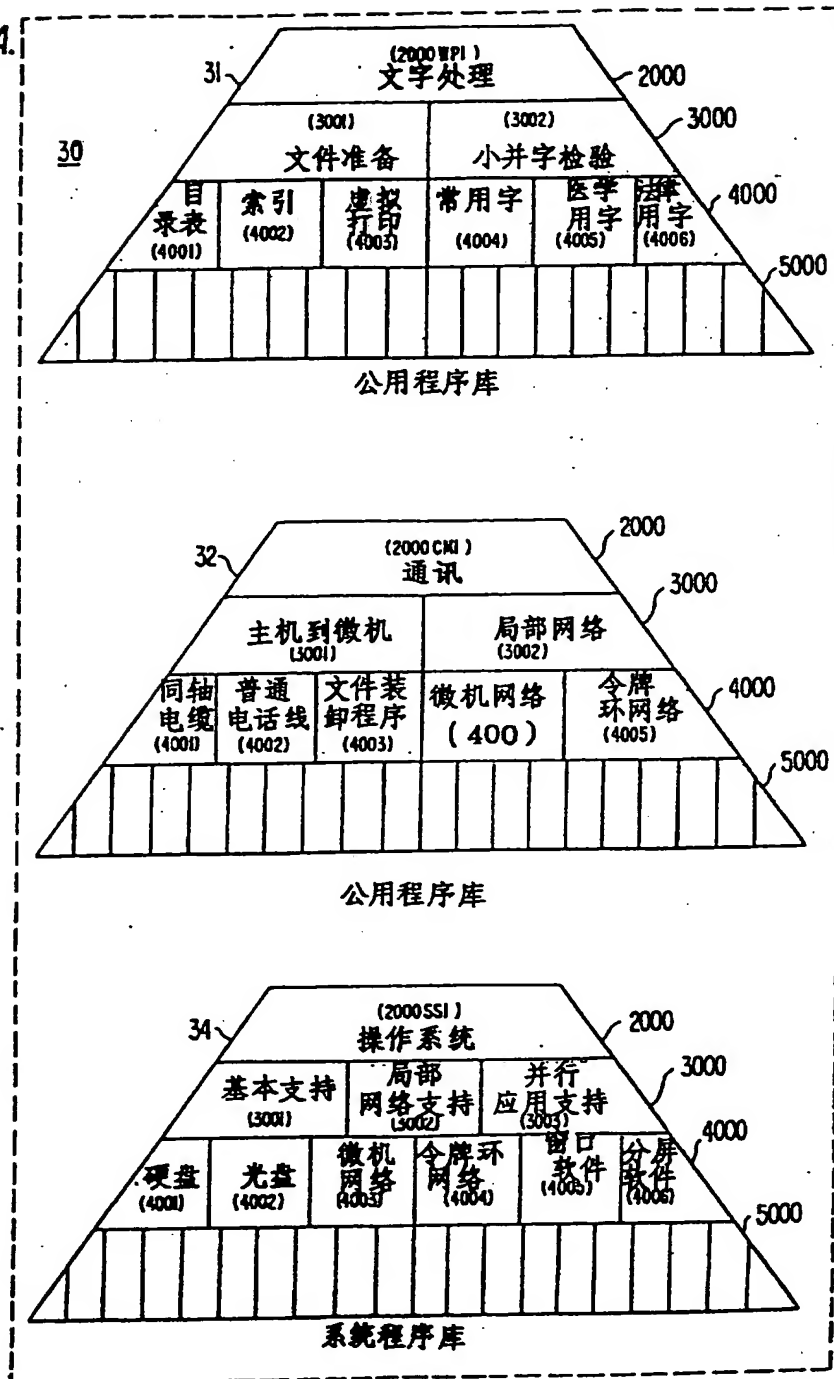


图 2B.

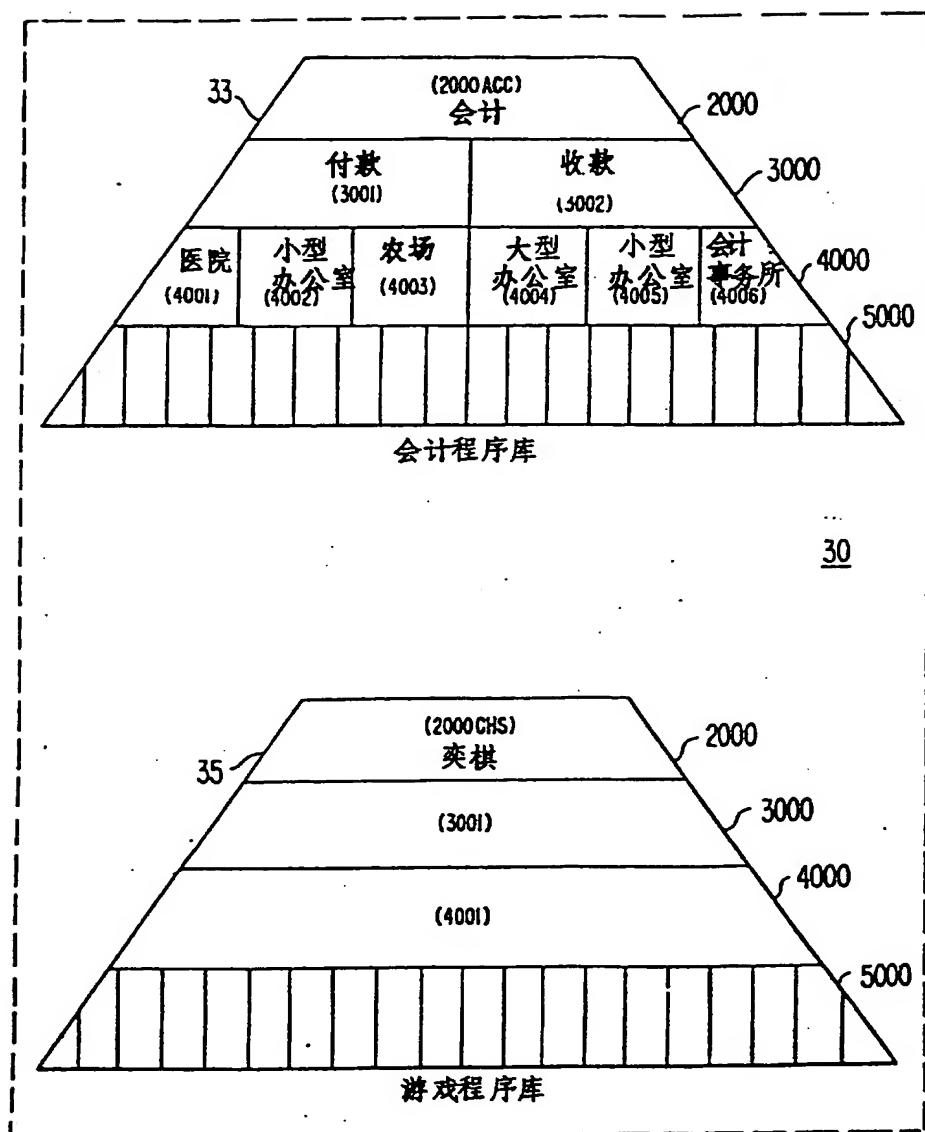


图 3.

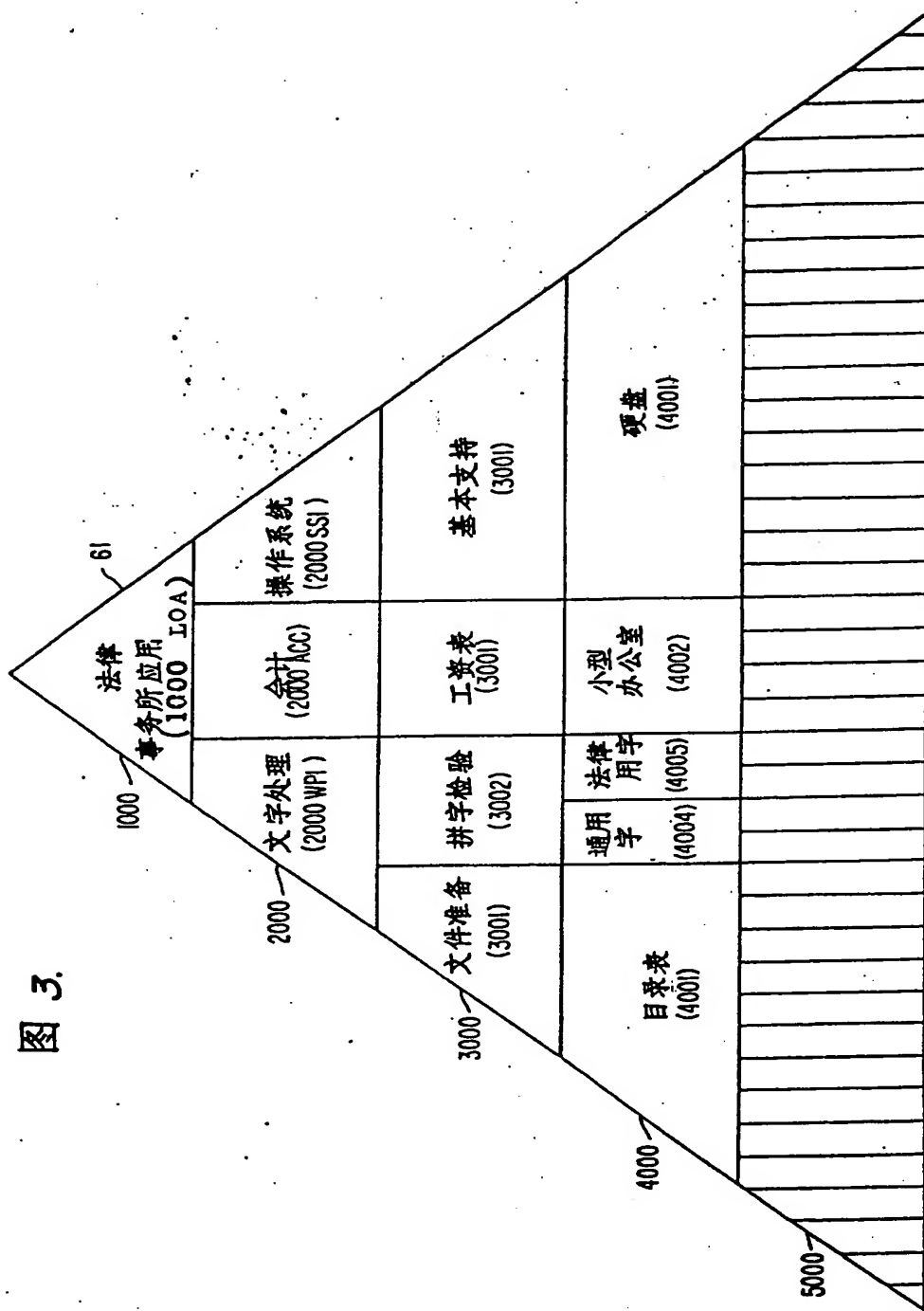


图 4.

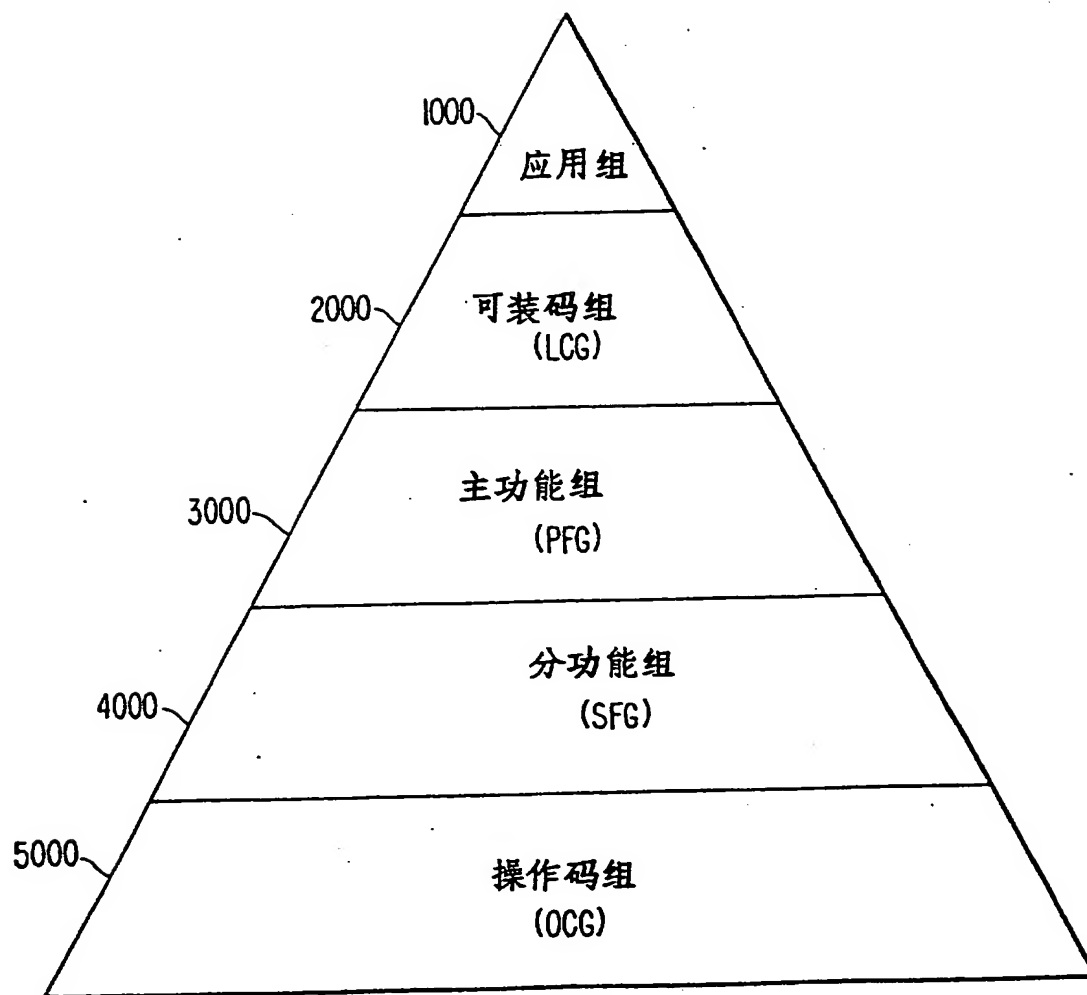


图 5.

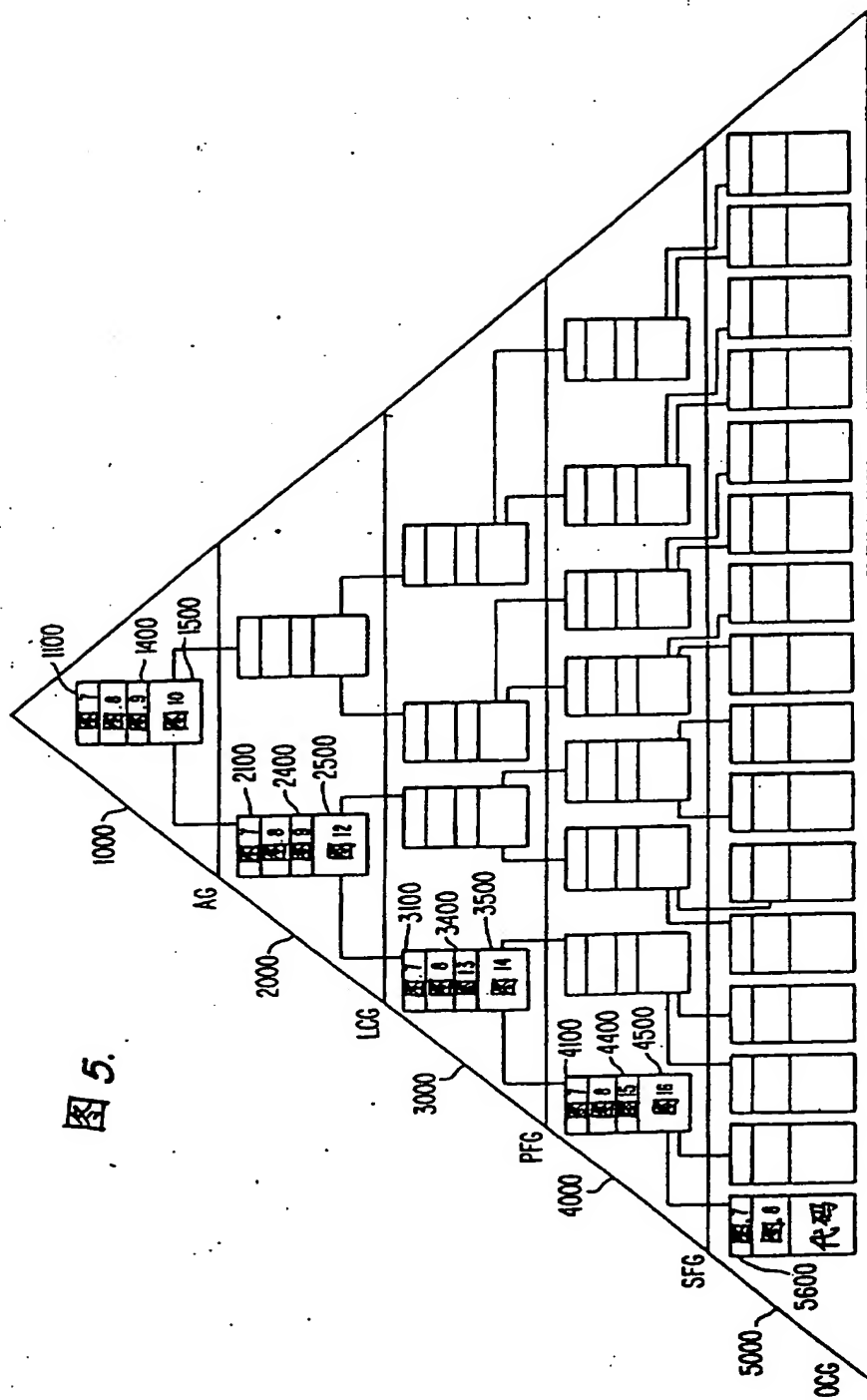


图 6A.

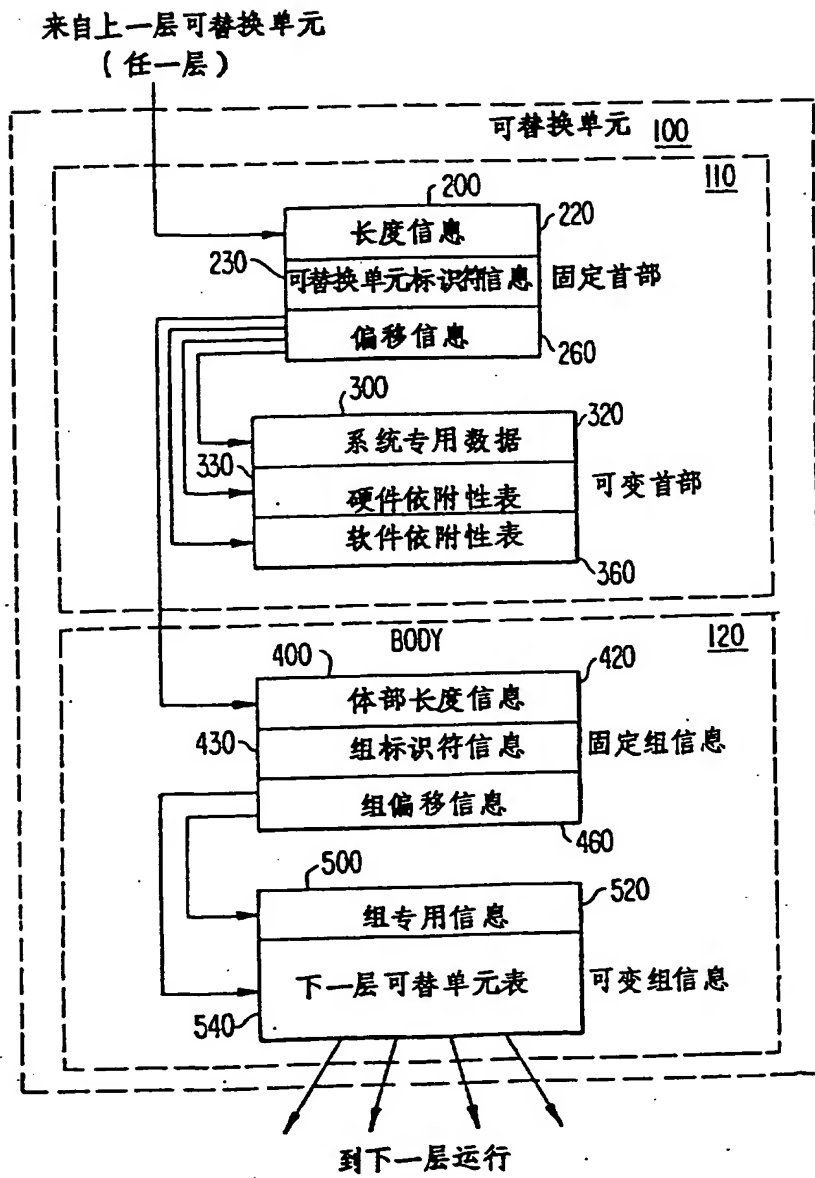


图 6B.

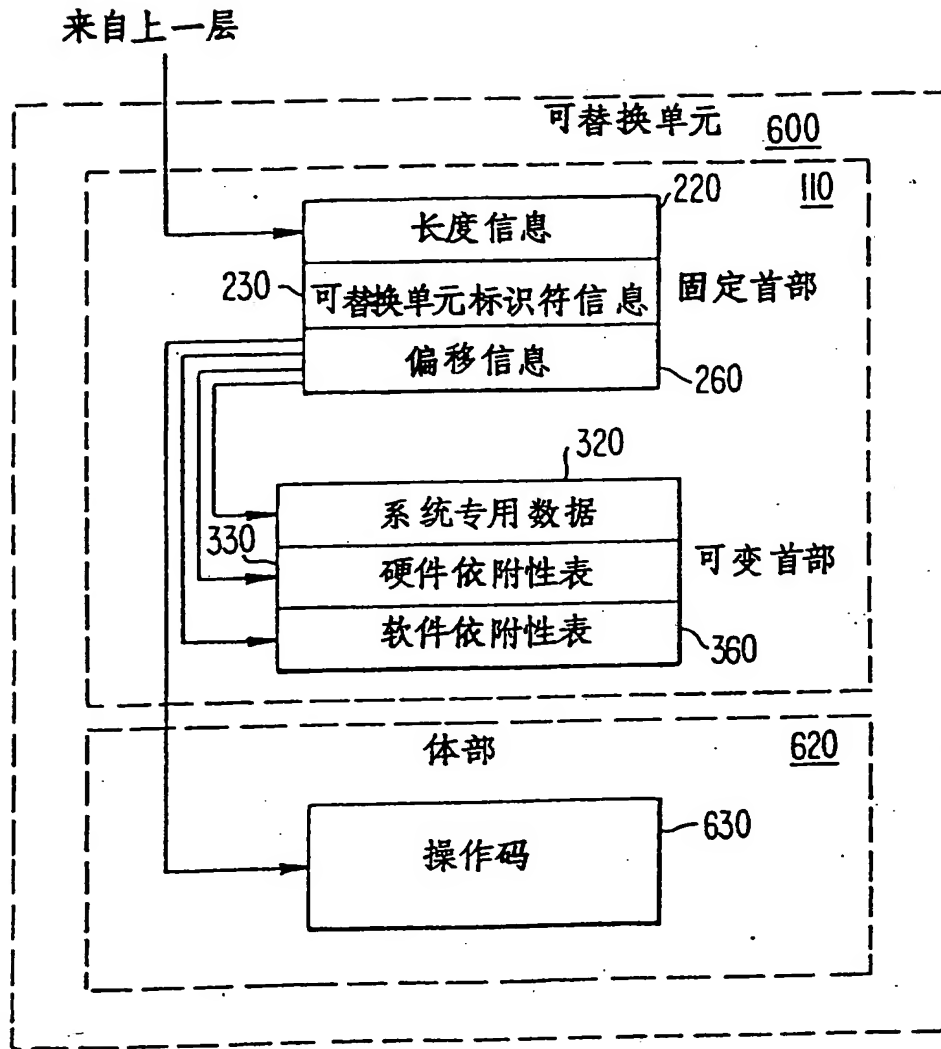


图 7
固定首部 200

可替换单元总长度	长度	名称	地址
体部长度	4	Y	B----221
保留	4	Y	B----222
单元名称	8	N	B----238
单元参照标识符	8	Y	C----231
单元装入标识符	4	Y	C----232
保留	8	N	C----233
可装码组或应用组标识符	1	N	C----239
主功能组标识符	7	Y	C----234
次功能组标识符	4	Y	C----236
单元指令系统	4	Y	C----237
单元控制信息	4	N	B----241
单元开发信息	4	Y	C----242
单元建立日期	4	Y	C----243
单元建立时间	8	Y	C----244
单元安装日期	6	Y	C----246
单元安装时间	8	Y	C----247
单元版本级	6	Y	C----248
单元发售/修订级	4	Y	C----249
保留	4	Y	C----250
单元临时固定标识符	1	N	C----251
保留	7	N	C----252
单元修补标识符	1	N	C----255
保留	7	N	C----253
单元编译程序名称	8	N	C----256
单元编译程序发售/修订级	10	N	C----257
单元公用名称	4	N	C----258
单元体部偏移	8	N	B----254
单元内重要产品数据偏移	4	Y	B----261
单元系统专用数据偏移	4	N	B----267
单元硬件依附表偏移	4	Y	B----262
单元软件依附表偏移	4	N	B----263
单元专用信息偏移	4	N	B----264
保留	4	N	B----265
存放单元的系统区域名	4	N	C----268
保留	8	N	B----266
单元修订日期	8	N	B----269
单元修订时间	8	N	C----270
首部扩展偏移	6	N	C----271
保留	4	N	B----272
	2	N	B----273

图 8

可变首部 300

	数 长	字 串	号 编
系统专用数据长度	4	Y	B----321
系统专用数据起点	4	Y	C----322
硬件依附表容量	4	Y	B----338
硬件依附表项目数	4	Y	B----331
保留	2	N	B----339
单元硬件相关标识符	2	N	B----332
硬件类型	4	Y	C----333
保留	1	N	B----334
硬件型号	3	N	C----335
硬件级装入标识符	4	N	C----336
硬件部品号	12	N	C----337
硬件依附表容量	4	Y	B----361
软件依附性表项目数	4	Y	B----362
项目容量	4	Y	B----363
软件相关标识符	2	N	C----364
保留	1	N	B----365
软件依附性类型	2	N	C----366
可装码组标识符	1	Y	C----367
可装码组版本级	4	Y	C----368
可装码组发售/修订级	4	Y	C----369
可装码组一主功能组标识符	4	N	B----370
可装码组一主功能组版本级	4	N	C----371
可装码组一主功能组发售/修订级	4	N	C----372
主功能组一分功能组标识符	4	N	B----373
主功能组一分功能组版本级	4	N	C----374
主功能组一分功能组发售/修订级	4	N	C----375
主功能组一分功能组程序改变级	4	N	C----376
首部扩展长度	4	N	B----381
单元部品号	12	N	C----382
单元简述	32	N	C----383
单元详述	60	N	C----384
单元专用信息长度	4	Y	B----385
单元专用信息起点	4	Y	C----386

图 9
固定应用组信息 1400

应用组信息	数据长度	数据类型	地址
应用组体部总长度	4	Y	B----1420
类主名	52	Y	C----1431
类主标识符	12	Y	C----1432
类主地址	52	N	C----1433
类主城市	32	N	C----1434
类主州/省	20	N	C----1435
类主国家	32	N	C----1436
类主邮政编码	20	N	C----1437
邮递员联络码	16	N	C----1438
类主电话号码	16	N	C----1439
应用组特征常数	4	N	C----1441
保留	5	N	B----1442
应用组标识符	7	Y	C----1443
应用组部品号	12	N	C----1444
应用组版本级	4	Y	C----1445
应用组发售/修订级	4	Y	C----1446
应用组中间程序变化级	4	Y	C----1447
应用组临时固定前级标识符	2	Y	C----1448
版权信息	64	Y	C----1449
保留	4	N	B----1461
保留	4	N	B----1462
可装码组表偏移	4	N	B----1463
应用组专用信息偏移	4	N	B----1464
保留	22	N	B----1465

图 10

可变应用组信息 1500

应用组信息	数据长度	数据类型	地址
应用组专用数据长度	4	N	B----1521
应用组专用数据起点	4	N	C----1522
可装码组表容量	4	Y	B----1541
保留	8	N	B----1542
应用组表中可装码组项目数	4	Y	B----1543
选择/删除	1	N	B----1545
可装码组可用性信息段	1	Y	C----1546
已装可装码组	1	N	C----1547
已用可装码组	1	N	C----1548
存放可装码组的系统地址	6	N	B----1549
保留	10	Y	C----1550
可装码组名称	6	N	B----1551
保留	10	Y	C----1552
可装码组标识符	5	N	B----1553
可装码组版本级	7	Y	C----1554
可装码组发售/修订级	4	Y	C----1555
	4	Y	C----1556

图 11.

固定可装码组信息 2400

	长度	字母	数据编号
可装码组体部总长度	4	Y	B----2420
卖主名	52	Y	C----2431
卖主标识符	12	Y	C----2432
卖主地址	52	N	C----2433
卖主城市	32	N	C----2434
卖主州/省	20	N	C----2435
卖主国家	32	N	C----2436
卖主邮政编码	20	N	C----2437
邮递员联络码	16	N	C----2438
卖主电话号码	16	N	C----2439
可装码组特征常数	4	N	C----2440
保留	5	N	B----2441
可装码组标识符	7	Y	C----2450
可装码组部品号	12	N	C----2451
可装码组版本级	4	Y	C----2452
可装码组发售/修订级	4	Y	C----2453
可装码组中间程序变化级	4	Y	C----2454
可装码组固定前缀标识符	2	Y	C----2455
版本信息	64	Y	C----2456
保留	4	N	B----2461
保留	4	N	B----2462
主功能表偏移	4	N	B----2463
可装码组专用信息偏移	4	N	B----2464
出口程序偏移	4	N	B----2467
应用组数据偏移	4	N	B----2465
保留	14	N	B----2466

图 12.

可变可装码组信息 2500

	长度	是否	段编号
可装码组专用数据长度	4	N	B----2521
可装码组专用数据起点	*	N	C----2522
应用组标识符	7	N	C----2531
应用组名称	16	N	C----2532
应用组地址	16	N	C----2533
主功能组表容量	4	Y	B----2541
保留	8	N	B----2542
表中项目数	4	Y	B----2543
选择/删除	1	N	B----2544
主功能组可用性段	1	Y	C----2545
已装主功能组	1	N	C----2546
已用主功能组的系统地址	1	N	C----2547
存放主功能组的系统地址	8	Y	C----2548
保留	8	N	B----2549
主功能组名称	8	Y	C----2550
保留	8	N	B----2551
主功能组标识符	4	Y	C----2552
主功能组版本级	4	Y	C----2553
主功能组发售/修订级	4	Y	C----2554
予存程序名称	16	N	C----2561
予存程序地址	16	N	C----2562
后存程序名称	16	N	C----2563
后存程序地址	16	N	C----2564
予恢复程序名称	16	N	C----2565
予恢复程序地址	16	N	C----2566
恢复程序名称	16	N	C----2567
恢复程序地址	16	N	C----2568
出口程序完成码	2	N	B----2569

图13

固定主功能组信息 3400

	数 长	存 取	中 继 段
主功能组体部总长度	4	Y	B----3420
主功能组特征常数	4	Y	C----3431
主功能组标识符	4	Y	C----3432
保留	5	N	B----3433
可装码组标识符	7	Y	C----3434
主功能组部品号	12	N	C----3435
主功能组版本级	4	Y	C----3436
主功能组发售/修订级	4	N	C----3437
主功能组中间程序变化组	4	Y	C----3438
主功能组临时固定前缀标识符	2	N	C----3439
版权信息	64	Y	C----3450
硬件依附性表偏移	4	N	B----3461
软件依附性表偏移	4	N	B----3462
次功能组表偏移	4	Y	B----3463
分功能组专用信息偏移	4	N	B----3464
保留	14	N	B----3465

图14

可变主功能组信息 3500

	数 长	存 取	中 继 段
主功能组专用数据长度	4	N	B----3521
主功能组专用数据起点	4	N	C----3522
分功能组表容量	4	Y	B----3541
保留	8	N	B----3542
表中项目数	4	Y	B----3543
选择/删除	1	N	B----3544
主功能组可能性段	1	Y	C----3545
已装分功能组	1	N	C----3546
已用分功能组	1	N	C----3547
存放分功能组的系统地址	8	Y	C----3548
保留	8	N	B----3549
分功能组名称	8	Y	C----3550
保留	8	N	B----3551
分功能组标识符	4	Y	C----3552
分功能组版本级	4	Y	C----3553
分功能组发售/修订级	4	Y	C----3554

图 15.

固定分功能组信息

4400

分功能组	体部	总长度	中	编
分功能组特征常数	4	4	Y	B-----4421
分功能组标识符	4	4	Y	C-----4422
保留	4	4	Y	C-----4431
分功能组一可装码组标识符	5	7	N	B-----4432
分功能组一主功能组标识符	4	12	Y	C-----4433
分功能组部品号	12		Y	B-----4434
分功能组版本级	4		N	C-----4435
分功能组发售/修订级	4		Y	C-----4436
分功能组中间程序变化级	4		N	C-----4437
分功能组临时固定前缀标识符	2	6	Y	C-----4438
版权信息	6		N	C-----4439
硬件依附性表偏移	4		Y	B-----4440
软件依附性表偏移	4		N	B-----4465
操作码组表偏移	4		N	B-----4466
分功能组临时固定/修补操作偏移	4		Y	B-----4461
分功能组专用信息偏移	4		N	B-----4462
保留	22		N	B-----4463
			N	B-----4464

图 1 6

可变分功能组信息

分功能组	专用数据长度	中	编
分功能组专用数据起点	4	Y	B-----4520
可装码组表容量	4	N	C-----4521
系统地址项目数	4	Y	B-----4541
系统地址表容量	4	Y	B-----4542
存放操作码组的系统地址	8	Y	B-----4543
保留	8	Y	C-----4544
操作码组表中项目数	4	N	B-----4545
操作码组名称	16	Y	B-----4546
操作码组副标识符	2	Y	C-----4547
操作码组版本级	4	N	B-----4548
操作码组发售/修订级	1	Y	C-----4549
保留	1	N	C-----4550
操作码组临时固定标识符	7	N	B-----4551
保留	1	N	C-----4552
操作码组修补标识符	7	N	B-----4553
临时固定/修补操作区	4	Y	C-----4554
修补操作偏移	4	N	B-----4561
已装临时固定项目数	1	N	B-----4562
保留	1	N	B-----4563
第一已装临时固定标识符	7	N	B-----4564
临时固定状态	4	N	C-----4565
已装修补项目数	1	N	B-----4566
保留	1	N	B-----4567
第一装修补标识符	7	N	B-----4568
修补状态	4	N	C-----4569
		N	B-----4570

图 17

程序设计师装配工具

11000 确定程序包

17000 显示程序包

图 32.

应用设计师装配工具

21000 确定应用程序包

24000 选择应用组功能

27000 显示应用程序包

图18.

确定程序包 11000
确定应用程序包 21000

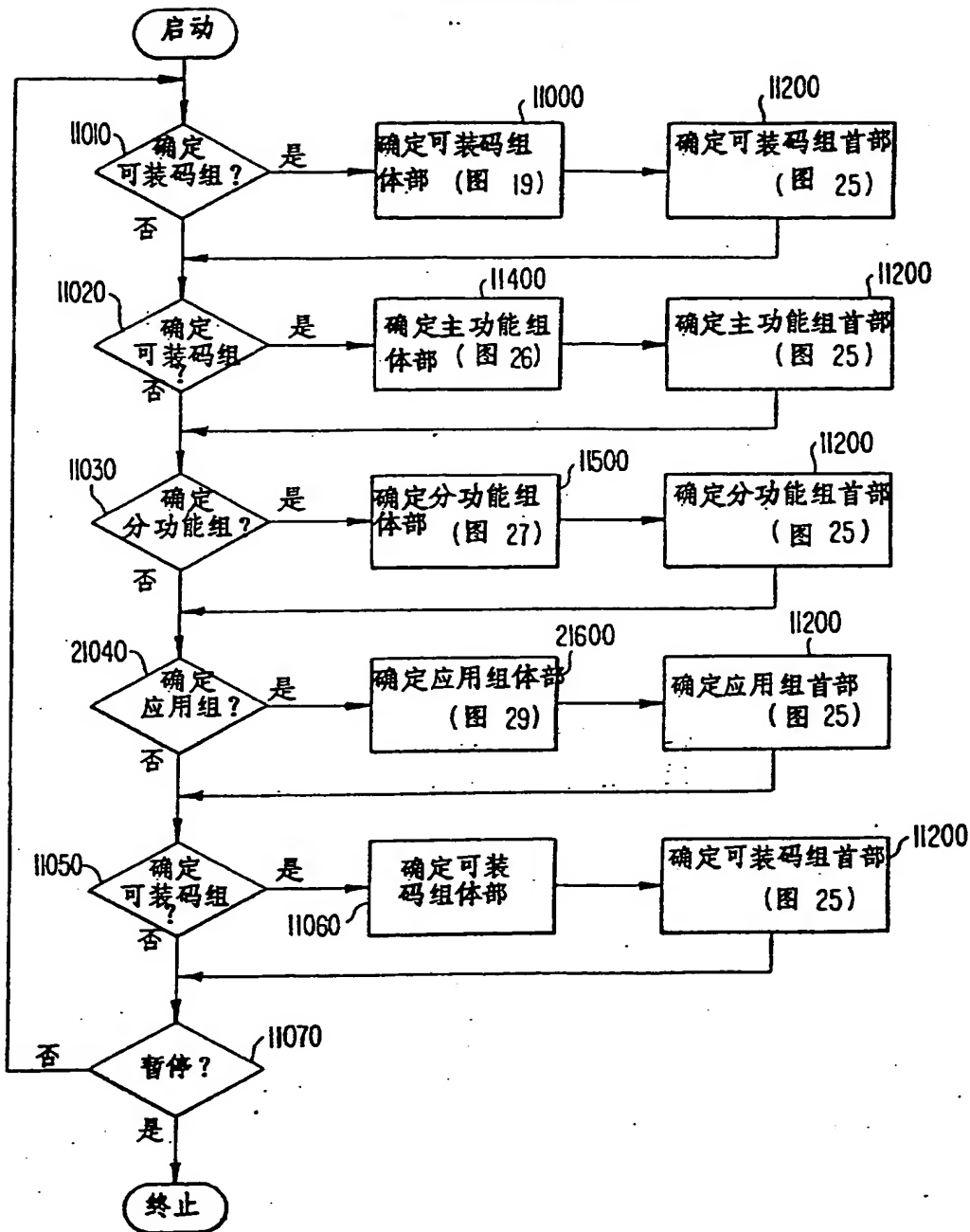


图 19.

确定可装码组体部 11100

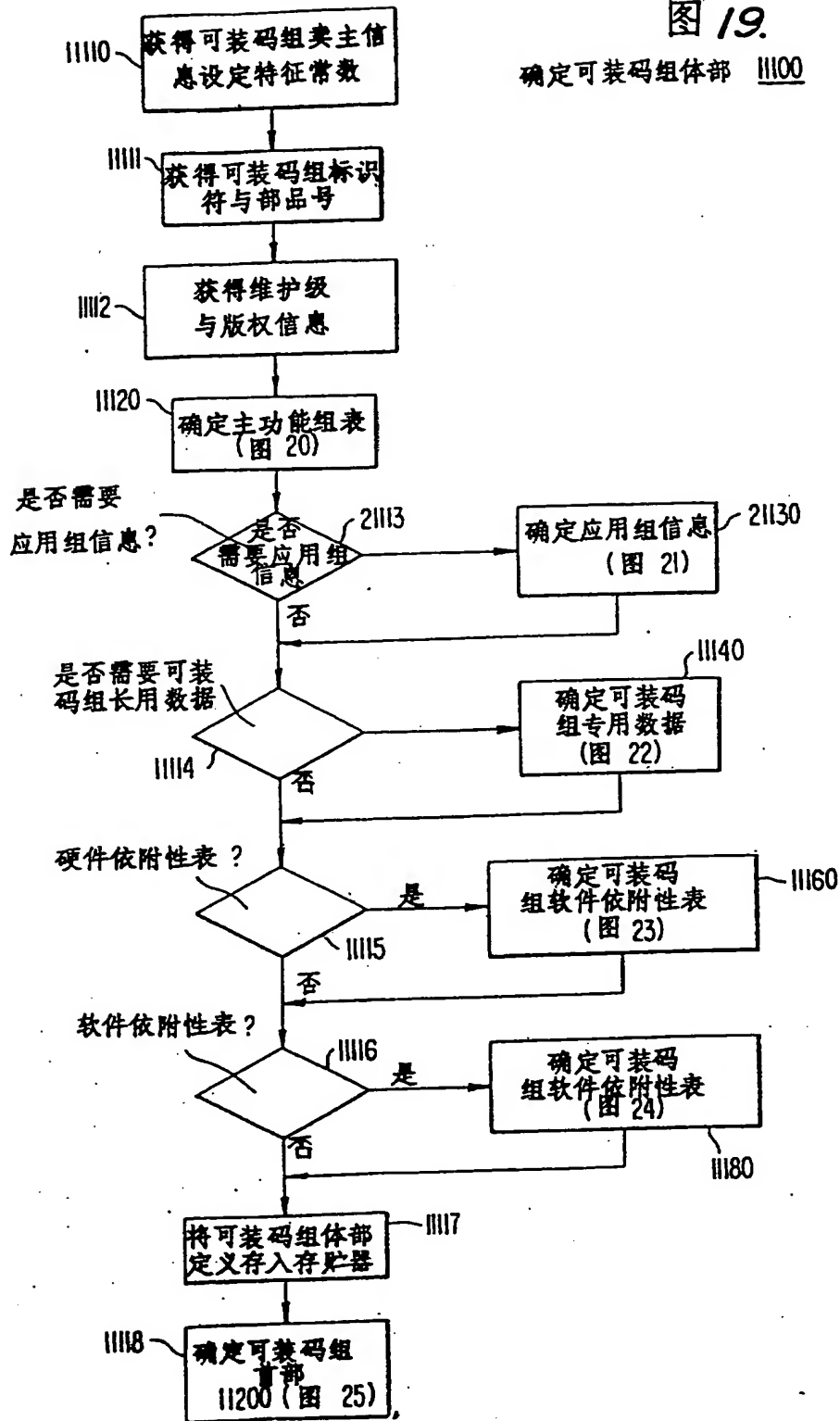


图 20.

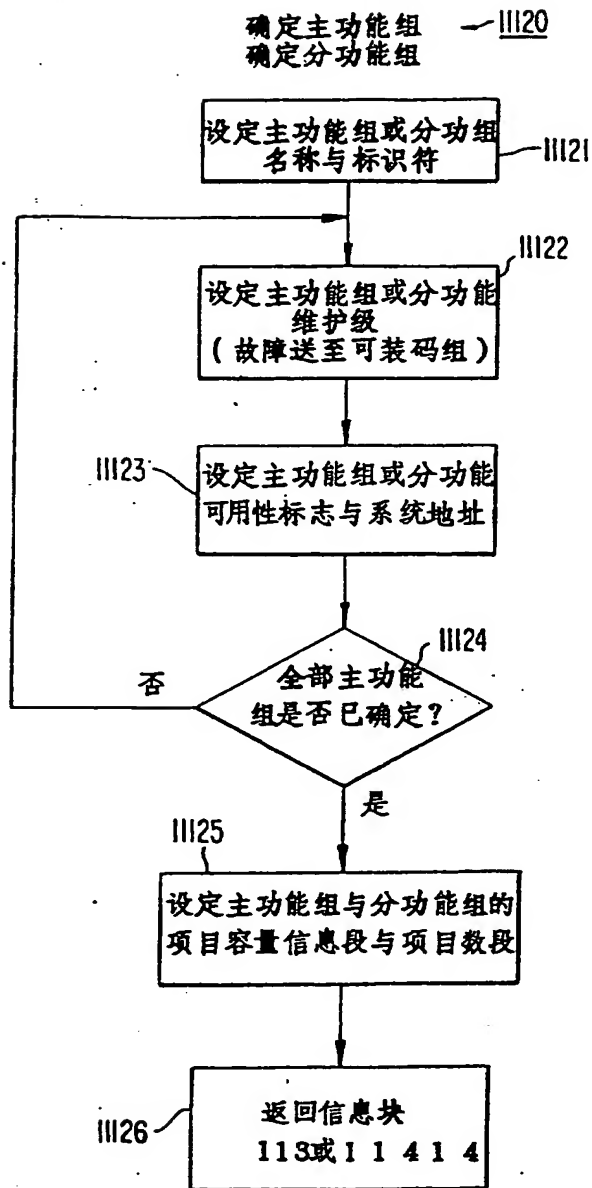


图 21.

确定应用组表 21130

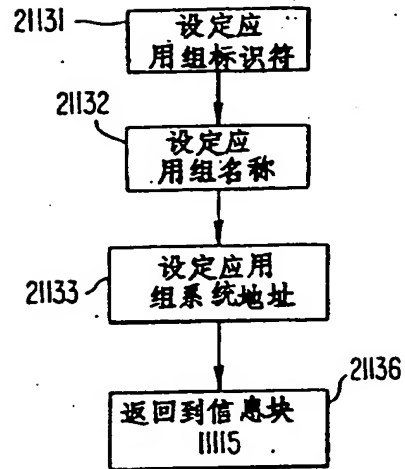


图 22.

确定应用组专用数据
确定可装码组专用数据
确定主功能组专用数据
确定分功能组专用数据

11140

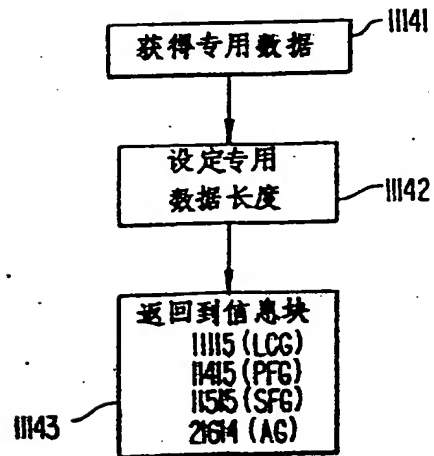


图 23.

确定可装码组硬件依附性表
 确定主功能组硬件依附性表
 确定分功能组硬件依附性表

III60

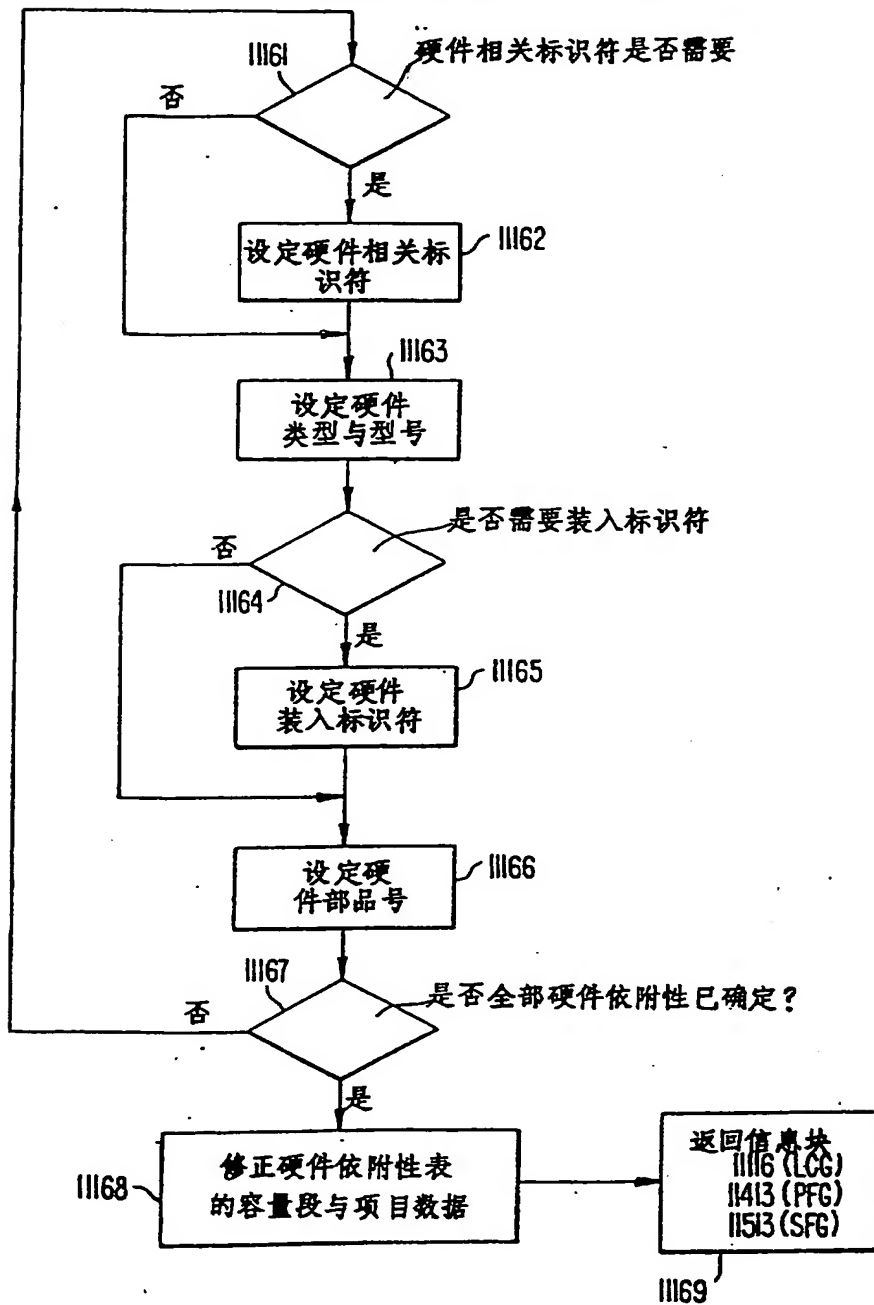


图 24

确定可装码组软件依附性表
 确定主功能组软件依附性表
 确定分功能组软件依附性表

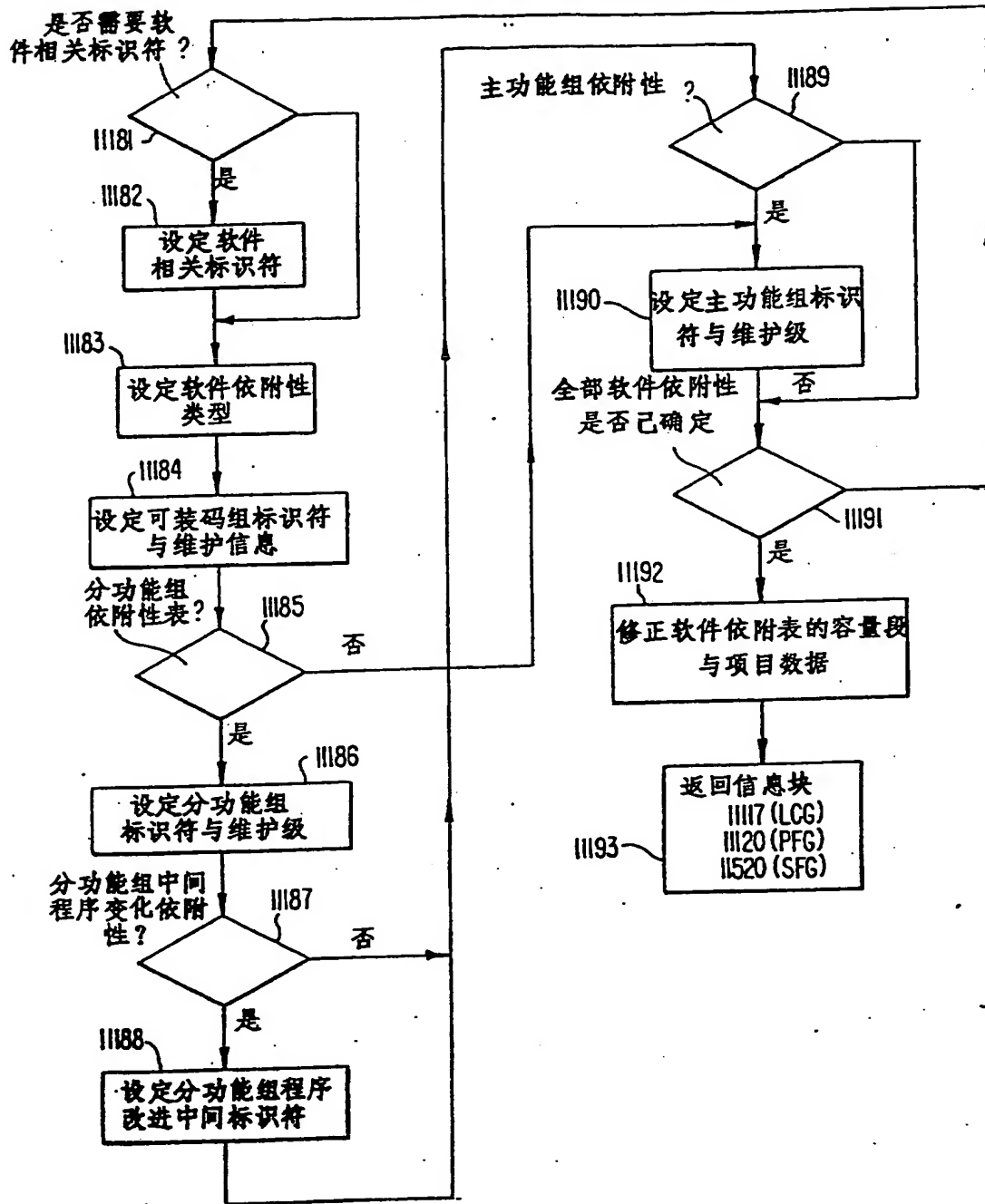


图 25A.

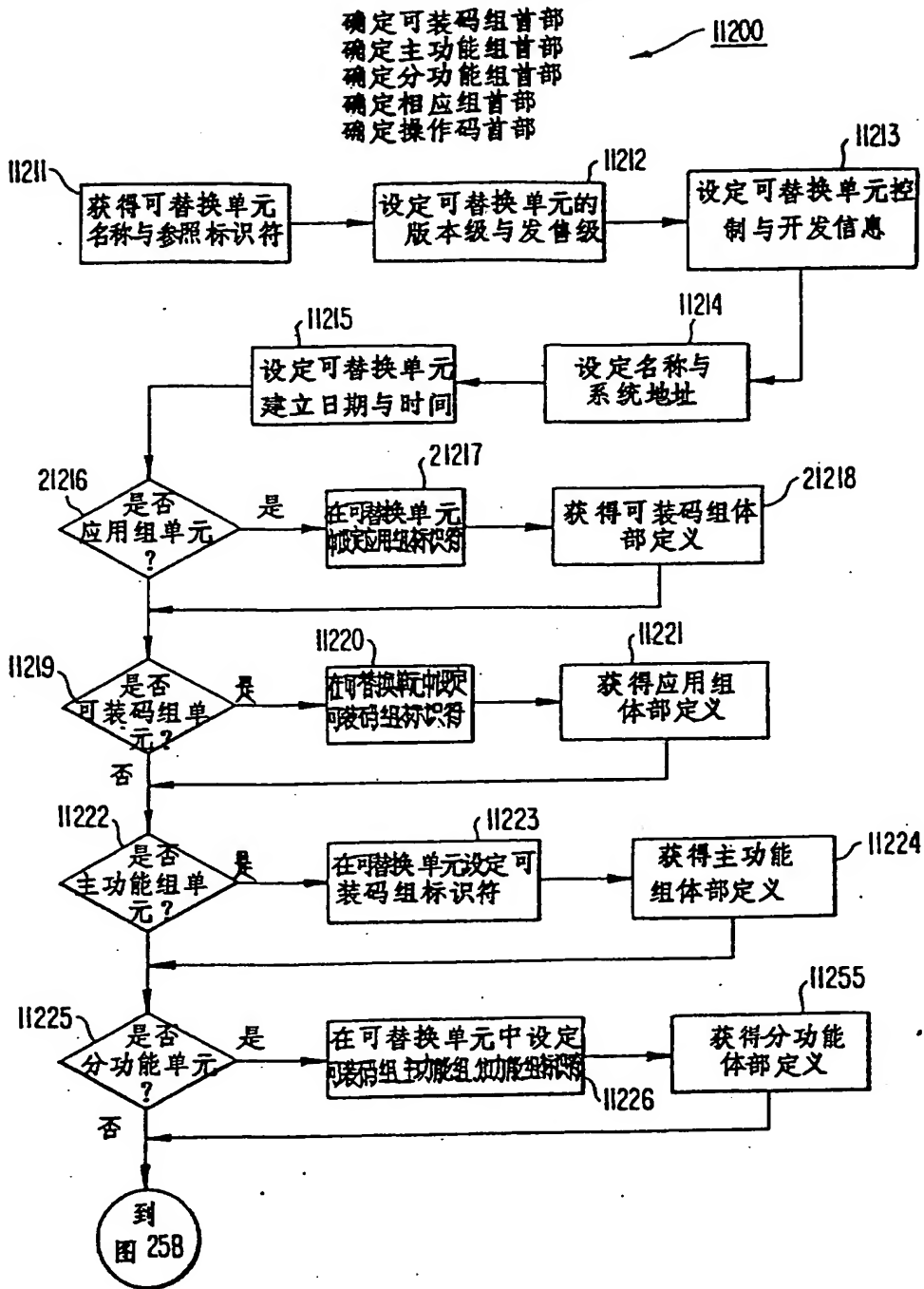


图 25B.

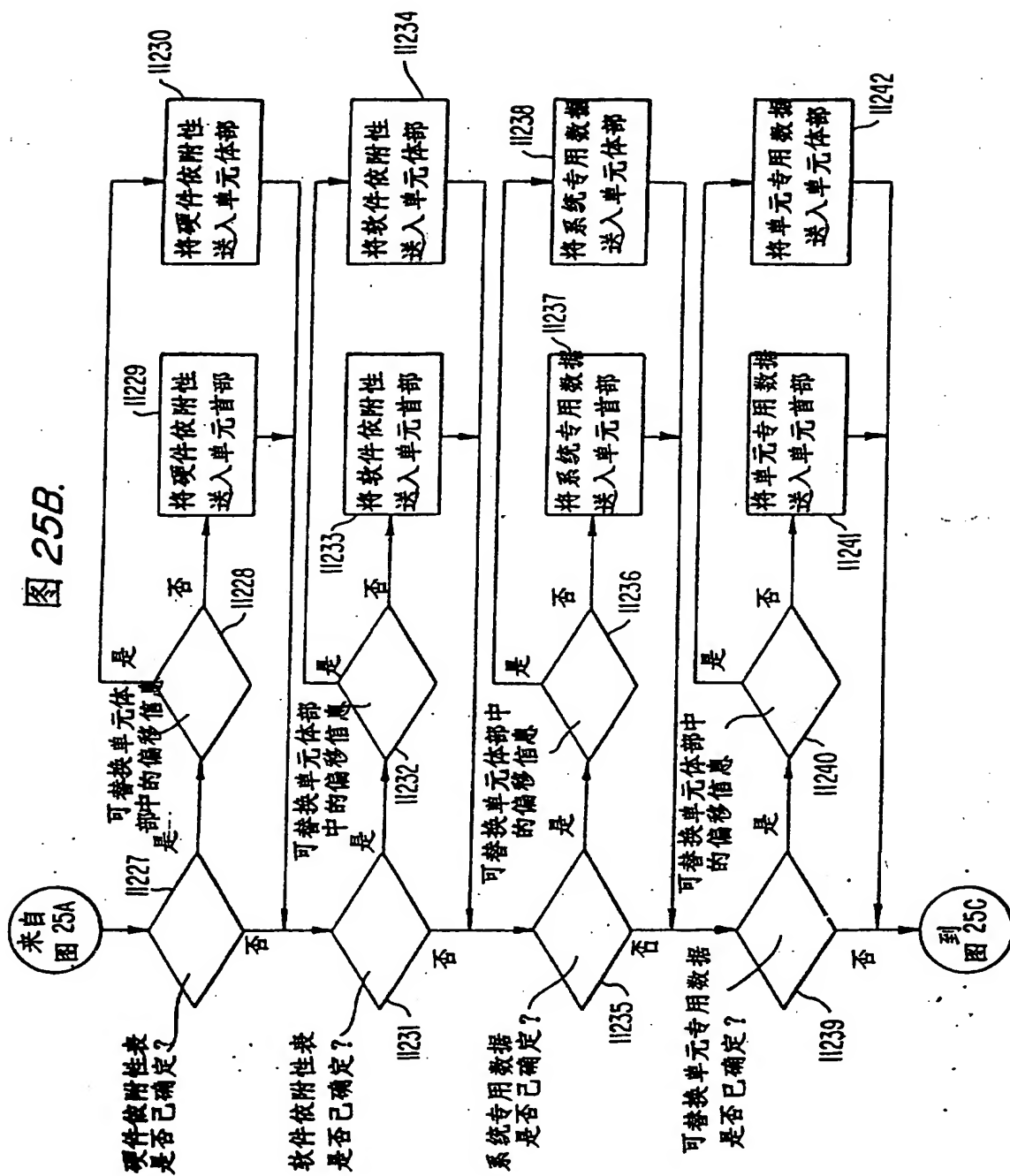


图 25C.

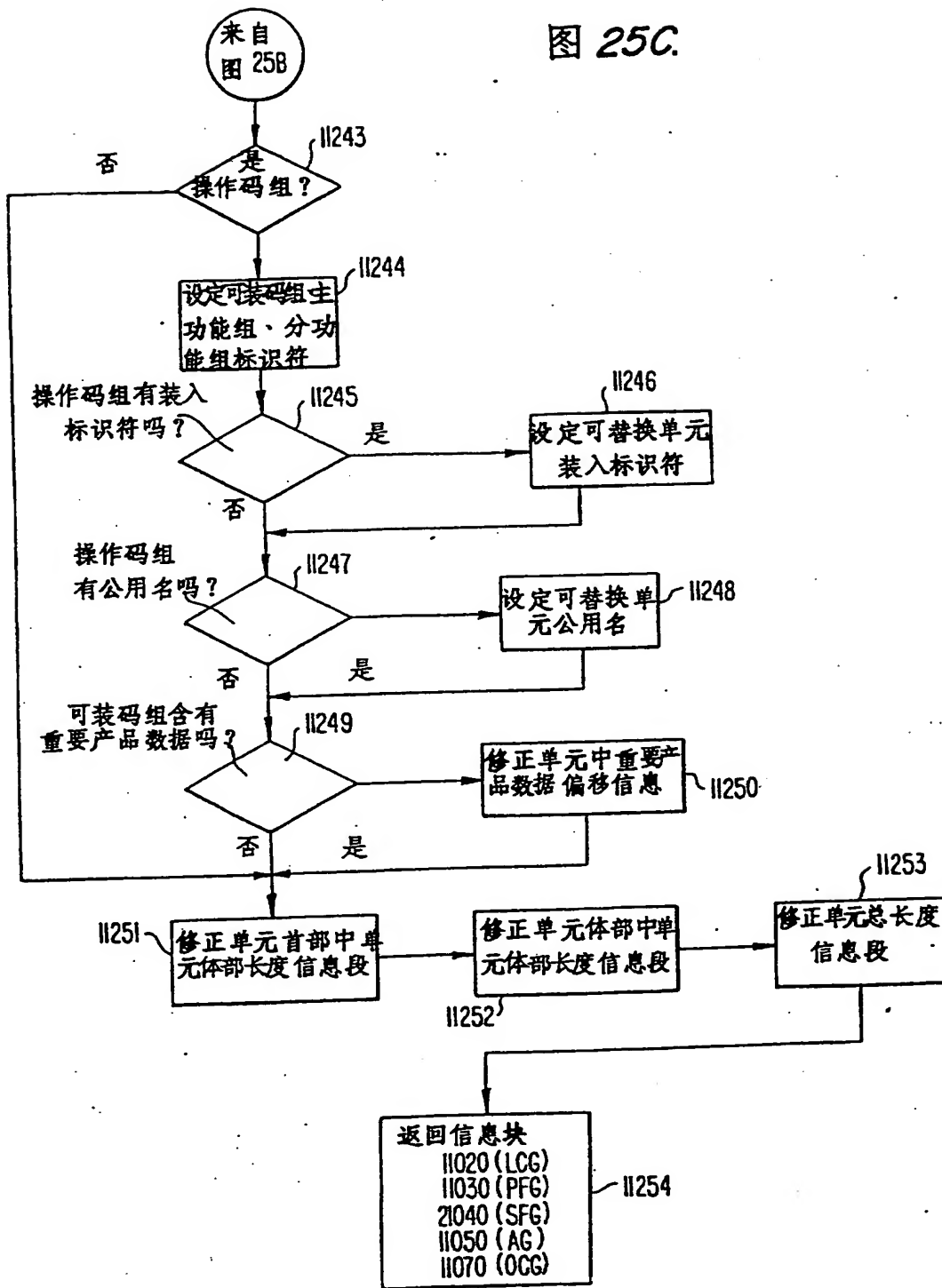


图 26.

确定主功能组体部 ← 11400

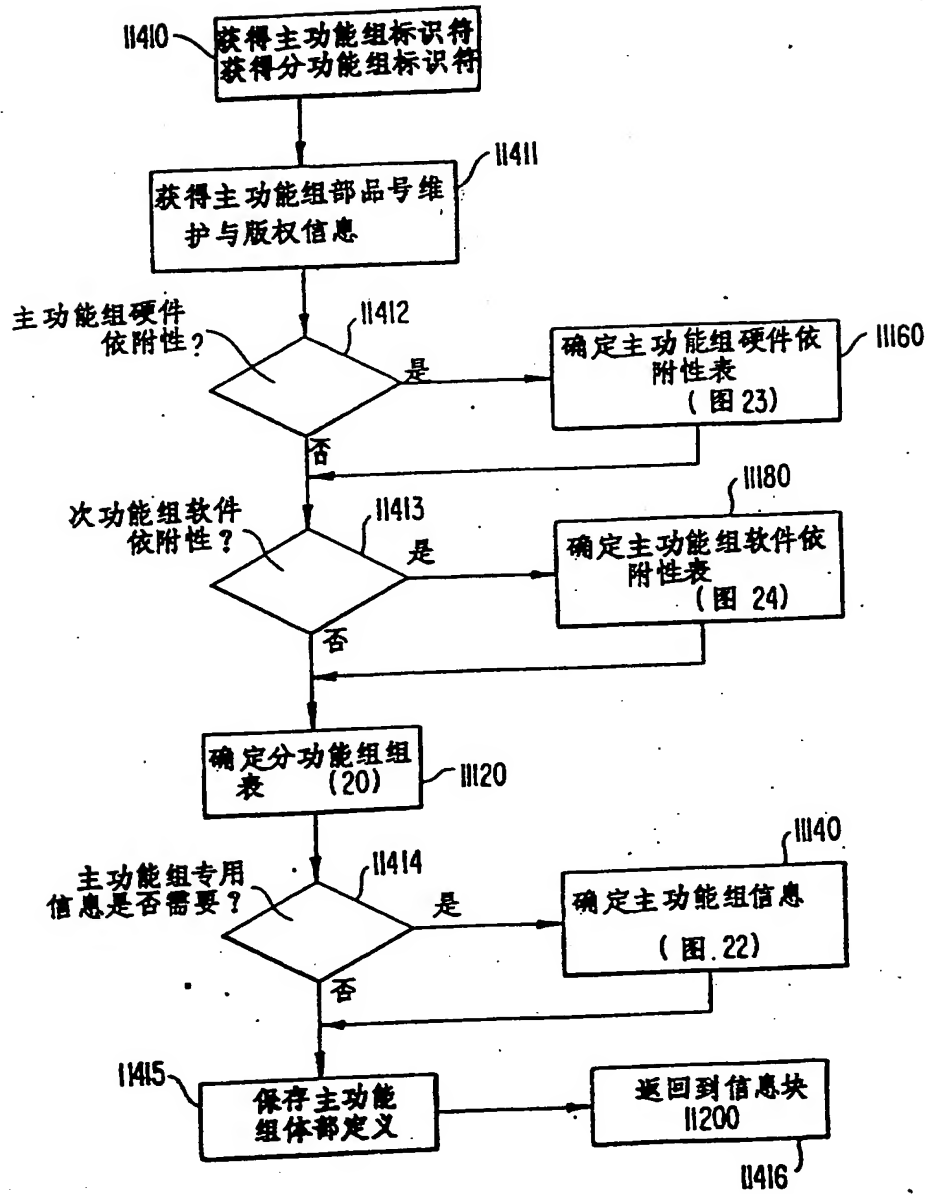


图 27

确定分功能组体部 11500

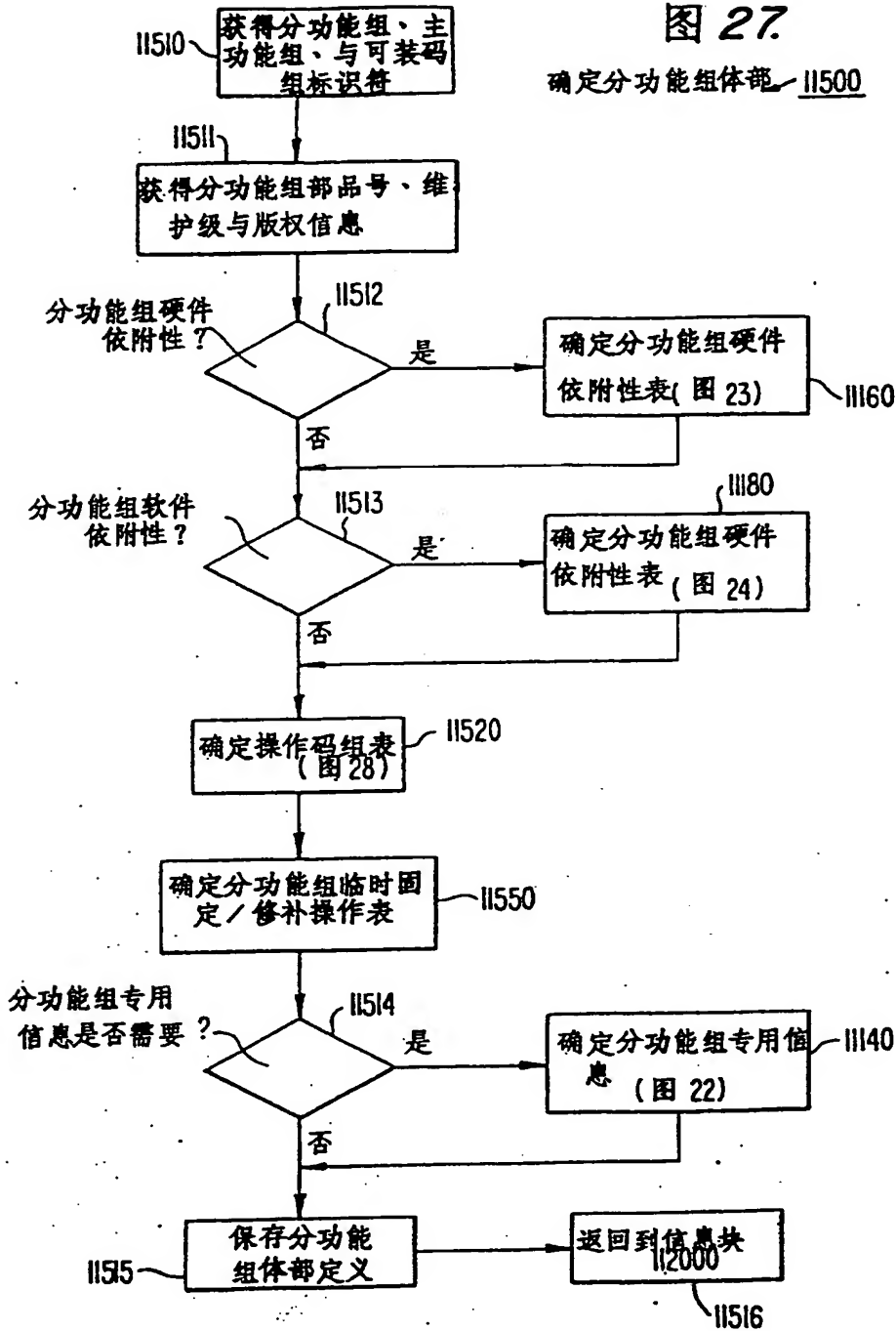


图 28.

确定操作码组表 11520

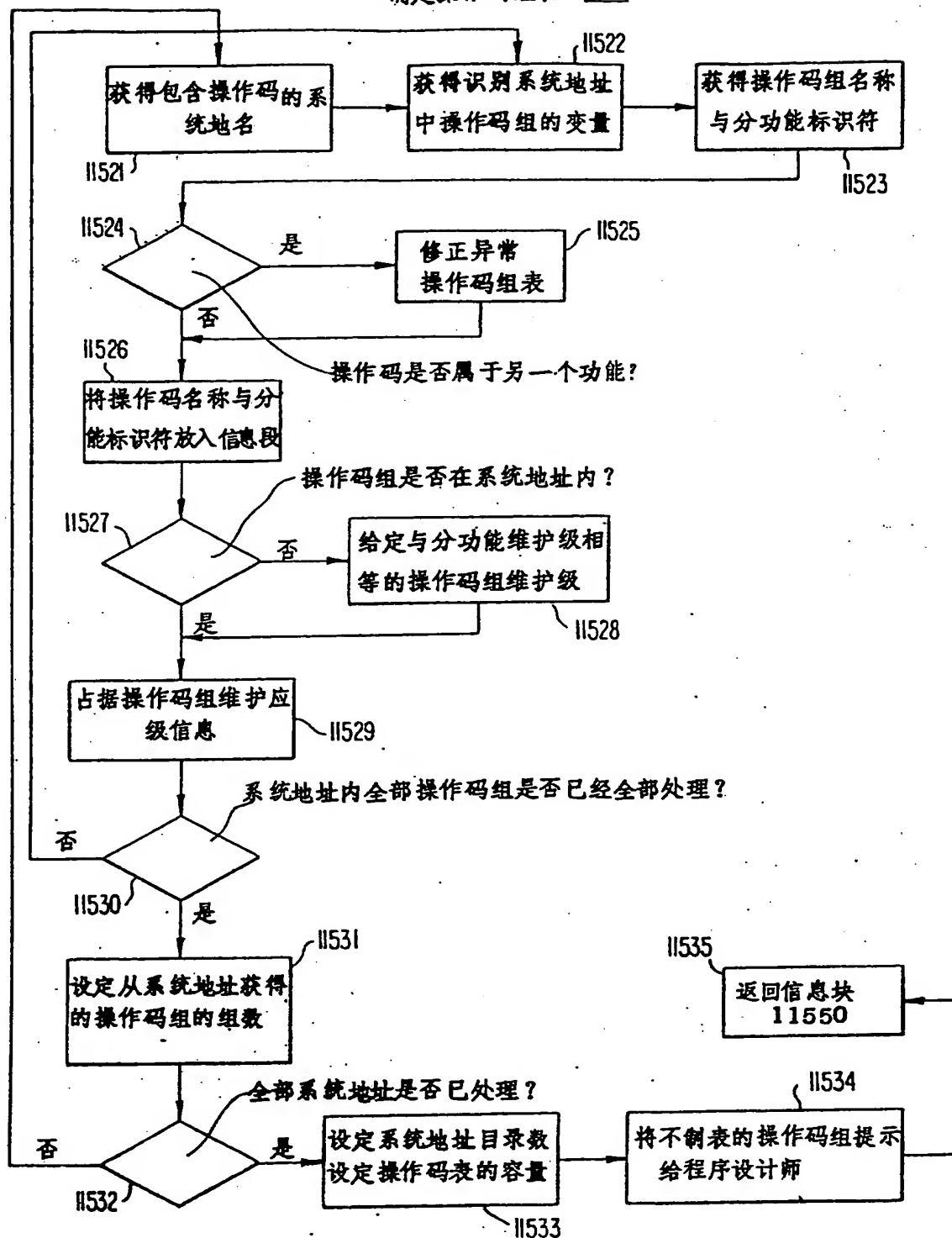


图 29.

确定应用组体部

21600

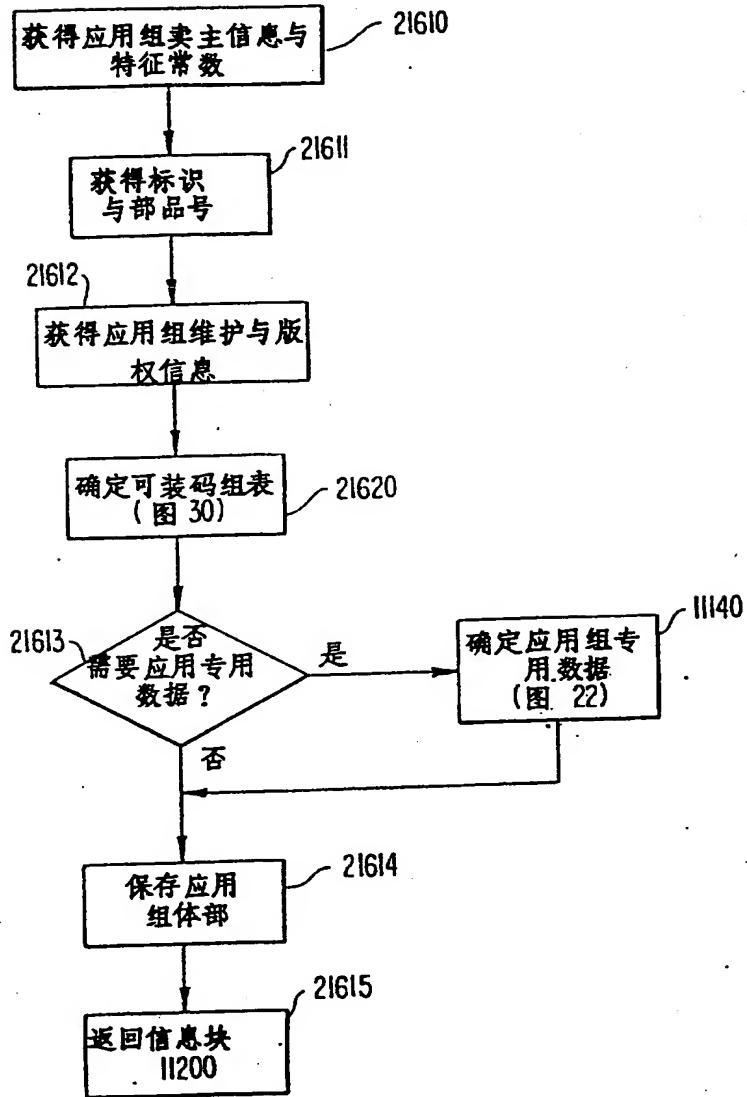


图 30.

确定可装码组 21620

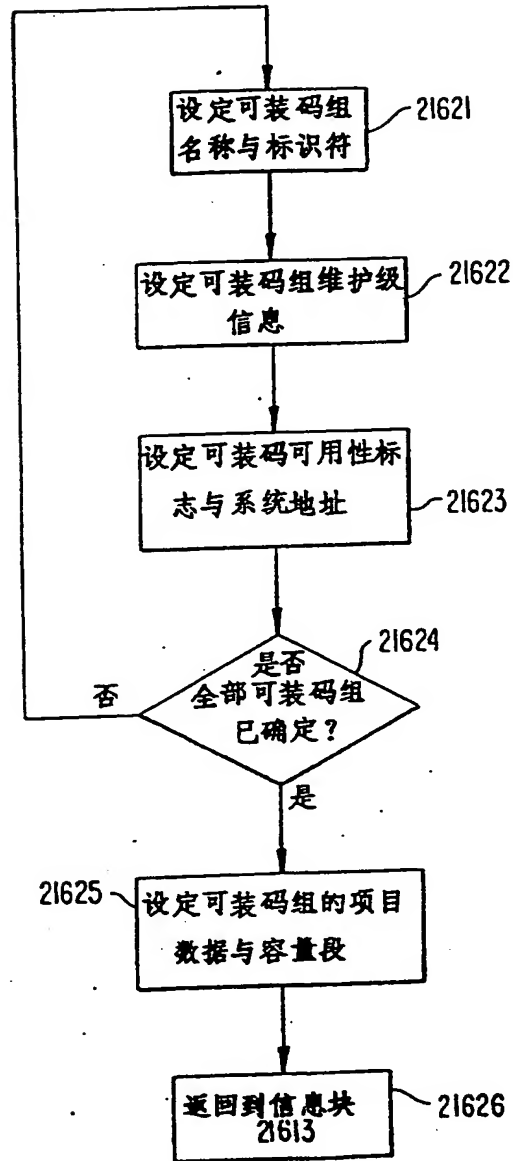


图 31.

显示程序包

17000

17050 DESCRIPTION	17100 LCG ID	17200 PFG ID	17300 SFG ID	17400 RU NAME	17500 RU TYPE	17600 LOCATION	17700 A	17800 I	17900 E
OPERATING SYSTEM	2000SS1			L2000SS1	LCG	LIBSYS	X	X	X
BASIC SUPPORT	2000SS1	3001		P3001SS1	PFG	LIBSYS	X	X	X
HARD DISK	2000SS1	3001	4001	S4001SS1	SFG	LIBSYS	X	X	X
OPTICAL DISK	2000SS1	3001	4002	S4002SS1	SFG	LIBSYS	X	X	X
LOCAL AREA NET	2000SS1	3002		P3002SS1	PFG	LIBLANS	X	X	X
PC NETWORK	2000SS1	3002	4003	S4003SS1	SFG	LIBLANS	X	X	X
TOKEN RING	2000SS1	3002	4004	S4004SS1	SFG		X		
COMMUNICATIONS	2000SS1	3003		P3003SS1	PFG	LIBCAS	X	X	
WINDOWING	2000SS1	3003	4005	S4005SS1	SFG		X		
SPLIT SCREEN	2000SS1	3001	4006	S4006SS1	SFG	LIBCAS	X	X	
COMMUNICATIONS	2000CM1			L2000CM1	LCG	LIBUTL	X	X	X
HOST TO PC	2000CM1	3001		P3001CM1	PFG	LIBUTL	X	X	X
COAXIAL CABLE	2000CM1	3001	4001	S4001CM1	SFG	LIBUTL	X	X	X
PHONE LINE	2000CM1	3001	4002	S4002CM1	SFG	LIBUTL	X	X	X
UP/DOWN LOAD	2000CM1	3001	4003	S4003CM1	SFG		X		
LOCAL AREA NET	2000CM1	3002		P3002CM1	PFG	LIBUTL	X	X	
PC NETWORK	2000CM1	3002	4004	S4004CM1	SFG	LIBUTL	X	X	
TOKEN RING	2000CM1	3002	4005	S4005CM1	SFG	LIBUTL	X	X	
WORD PROCESSING	2000WP1			L2000WP1	LCG	LIBUTL	X	X	X
DOCUMENT PREP	2000WP1	3001		P3001WP1	PFG	LIBUTL	X	X	X
TABLE OF CONTENT	2000WP1	3001	4001	S4001WP1	SFG	LIBUTL	X	X	X
INDEX	2000WP1	3001	4002	S4002WP1	SFG	LIBUTL	X	X	X
FANCY PRINTING	2000WP1	3001	4003	S4003WP1	SFG	LIBUTL	X	X	X
SPELL CHECKER	2000WP1	3002		P3002WP1	PFG	LIBUTL	X	X	X
GENERAL WORDS	2000WP1	3002	4004	S4004WP1	SFG	LIBUTL	X	X	X
MEDICAL WORDS	2000WP1	3002	4005	S4005WP1	SFG	LIBUTL	X	X	X
LEGAL WORDS	2000WP1	3002	4006	S4006WP1	SFG	LIBUTL	X	X	X
ACCOUNTING	2000ACC			L2000ACC	LCG	LIBACC	X	X	X
PAYROLL	2000ACC	3001		P3001ACC	PFG	LIBACC	X	X	X
HOSPITAL	2000ACC	3001	4001	S4001ACC	SFG		X		
SMALL OFFICE	2000ACC	3001	4002	S4002ACC	SFG	LIBACC	X	X	X
FARM	2000ACC	3001	4003	S4003ACC	SFG		X		
ACCOUNTS RCYBLE	2000ACC	3002		P3002ACC	PFG	LIBACC	X	X	X
LARGE OFFICE	2000ACC	3002	4004	S4004ACC	SFG	LIBACC	X	X	X
SMALL OFFICE	2000ACC	3002	4005	S4005ACC	SFG	LIBACC	X	X	X
ACCTG FIRM	2000ACC	3002	4006	S4006ACC	SFG	LIBACC	X	X	X
CHESS	2000CHS			L2000CHS	LCG	LIBGAME	X	X	X
CHESS FUNCTION	2000CHS	3001		P3001CHS	PFG	LIBGAME	X	X	X
CHESS LOAD	2000CHS	3001	4001	S4001CHS	SFG	LIBGAME	X	X	X

图 33.

选择应用组功能 24000

24010 S	24050 DESCRIPTION	24100 LCG ID	24200 PFG ID	24300 SFG ID	24400 RU NAME	24500 RU TYPE	24600 LOCATION	24700 A	24800 I	24900 C
	LEGAL OFCE APPLN	1000AG1			L1001AG1	AG	LIBGBL			
X	OPERATING SYSTEM	2000SS1			L2000SS1	LCG	LIBSYS	X	X	X
X	BASIC SUPPORT	2000SS1	3001		P3001SS1	PFG	LIBSYS	X	X	X
X	HARD DISK	2000SS1	3001	4001	S4001SS1	SFG	LIBSYS	X	X	X
	OPTICAL DISK	2000SS1	3001	4002	S4002SS1	SFG	LIBSYS	X	X	X
	LOCAL AREA NET	2000SS1	3002		P3002SS1	PFG	LIBLANS	X	X	X
	PC NETWORK	2000SS1	3002	4003	S4003SS1	SFG	LIBLANS	X	X	X
	TOKEN RING	2000SS1	3002	4004	S4004SS1	SFG		X	X	
	COMMUNICATIONS	2000SS1	3003		P3003SS1	PFG	LIBCAS	X	X	
	WINDOWING	2000SS1	3003	4005	S4005SS1	SFG		X	X	
	SPLIT SCREEN	2000SS1	3001	4006	S4006SS1	SFG	LIBCAS	X	X	
X	WORD PROCESSING	2000WP1			L2000WP1	LCG	LIBUTL	X	X	X
X	DOCUMENT PREP	2000WP1	3001		P3001WP1	PFG	LIBUTL	X	X	X
X	TABLE OF CONTENT	2000WP1	3001	4001	S4001WP1	SFG	LIBUTL	X	X	X
	INDEX	2000WP1	3001	4002	S4002WP1	SFG	LIBUTL	X	X	X
	FANCY PRINTING	2000WP1	3001	4003	S4003WP1	SFG	LIBUTL	X	X	X
X	SPELL CHECKER	2000WP1	3002		P3002WP1	PFG	LIBUTL	X	X	X
X	GENERAL WORDS	2000WP1	3002	4004	S4004WP1	SFG	LIBUTL	X	X	X
X	MEDICAL WORDS	2000WP1	3002	4005	S4005WP1	SFG	LIBUTL	X	X	X
X	LEGAL WORDS	2000WP1	3002	4006	S4006WP1	SFG	LIBUTL	X	X	X
X	ACCOUNTING	2000ACC			L2000ACC	LCG	LIBACC	X	X	X
X	PAYROLL	2000ACC	3001		P3001ACC	PFG	LIBACC	X	X	X
	HOSPITAL	2000ACC	3001	4001	S4001ACC	SFG		X	X	X
X	SMALL OFFICE	2000ACC	3001	4002	S4002ACC	SFG	LIBACC	X	X	X
	FARM	2000ACC	3001	4003	S4003ACC	SFG		X	X	X
	ACCOUNTS RCYBLE	2000ACC	3002		P3002ACC	PFG	LIBACC	X	X	X
	LARGE OFFICE	2000ACC	3002	4004	S4004ACC	SFG	LIBACC	X	X	X
	SMALL OFFICE	2000ACC	3002	4005	S4005ACC	SFG	LIBACC	X	X	X
	ACCTG FIRM	2000ACC	3002	4006	S4006ACC	SFG	LIBACC	X	X	X

图 34.

显示应用程序包 27000

27010	27050	27100	27200	27300	27400	27500	27600	27700	27800	27900
S	DESCRIPTION	LCG ID	PFG ID	SFG ID	RU NAME	RU TYPE	LOCATION	A	I	E
	LEGAL OFCE APPLN	1000AG1			L1001AG1	AG	LIBGBL			
X	OPERATING SYSTEM	2000SSI			L2000SSI	LCG	LIBSYS	X	X	X
X	BASIC SUPPORT	2000SSI	3001		P3001SSI	PFG	LIBSYS	X	X	X
X	HARD DISK	2000SSI	3001	4001	S4001SSI	SFG	LIBSYS	X	X	X
X	WORD PROCESSING	2000WPI			L2000WPI	LCG	LIBUTL	X	X	X
X	DOCUMENT PREP	2000WPI	3001		P3001WPI	PFG	LIBUTL	X	X	X
X	TABLE OF CONTENT	2000WPI	3001	4001	S4001WPI	SFG	LIBUTL	X	X	X
X	SPELL CHECKER	2000WPI	3002		P3002WPI	PFG	LIBUTL	X	X	X
X	GENERAL WORDS	2000WPI	3002	4004	S4004WPI	SFG	LIBUTL	X	X	X
X	LEGAL WORDS	2000WPI	3002	4006	S4006WPI	SFG	LIBUTL	X	X	X
X	ACCOUNTING	2000ACC			L2000ACC	LCG	LIBACC	X	X	X
X	PAYROLL	2000ACC	3001		P3001ACC	PFG	LIBACC	X	X	X
X	SMALL OFFICE	2000ACC	3001	4002	S4002ACC	SFG	LIBACC	X	X	X

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox